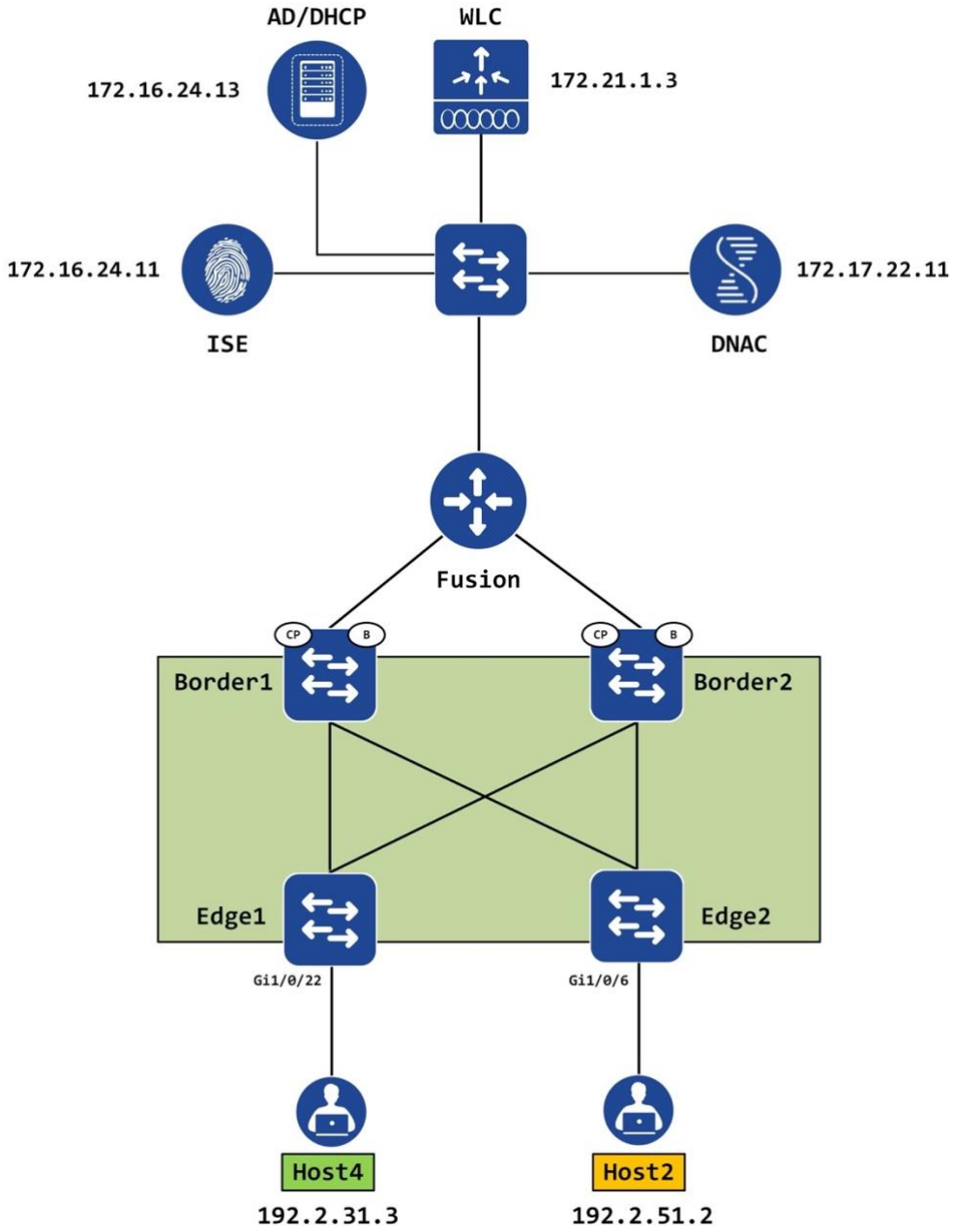
Topology
Head-end replication
Configuring head-end replication via DNAC
Source and Receiver in fabric
Building the overlay shared tree
Start of source stream
Building the source tree
SPT cutover
Decoding and understanding PIM join attributes for LISP
Final MRIB state on each device
Source in fabric, receiver outside fabric
Building the shared tree
Start of source stream
Building the source tree
SPT cutover
Final MRIB state on each device
Source in fabric, receiver in fabric and outside fabric
Building the shared tree
Start of source stream
Building the source tree
SPT cutover
Final MRIB state on each device
Implications on IP adjacency usage
Native Multicast
ASM in SDA
Configuring ASM in the overlay via DNAC
Source and Receiver in fabric
Building the shared tree
Start of source stream
Building the source tree
SPT cutover
Final MRIB state on each device
ASM with dual RPs
Configuring dual RPs via DNAC
Receiver join and building of shared tree
Source start and building the SPT
Understanding the MSDP process
SSM in SDA
Configuring SSM in the overlay via DNAC
Source and Receiver in fabric
Start of source stream
Receiver join and building the SPT
Final MRIB state and packet flow
Source in fabric and Receiver outside fabric
Receiver join and building the SPT
Final MRIB state and packet flow
Source in fabric, Receiver in and outside fabric
Receiver join and building SPT
Final MRIB state and packet flow

Aninda Chatterjee (aninchat@cisco.com)

Topology

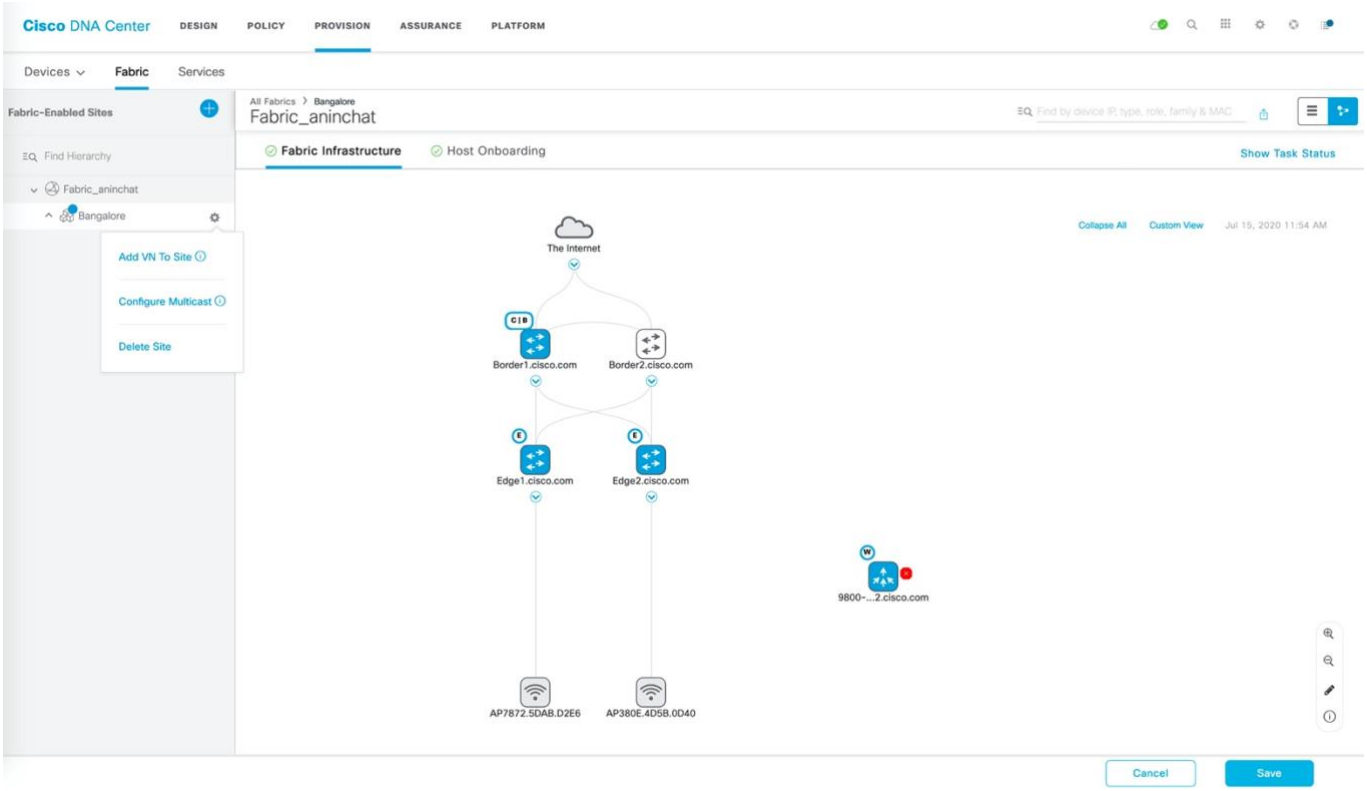


Aninda Chatterjee (aninchat@cisco.com)

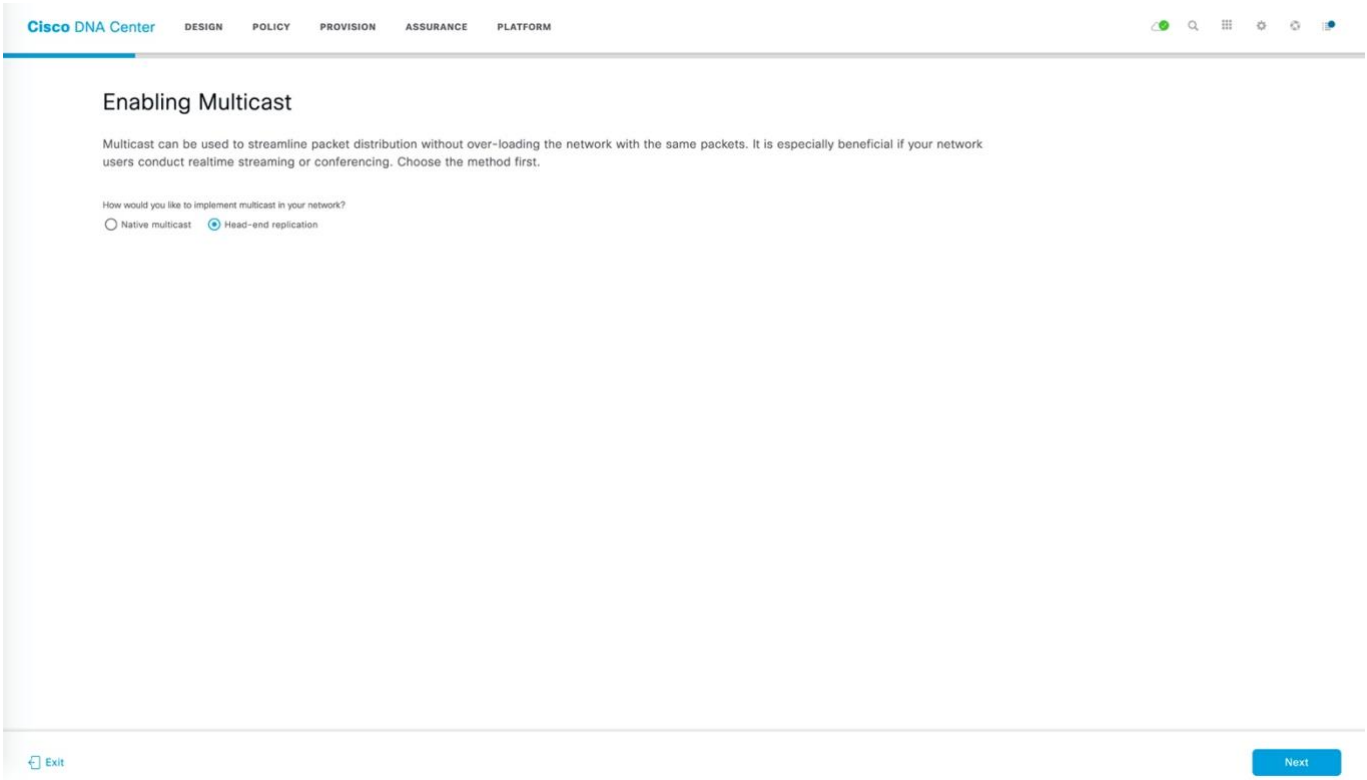
Head-end replication in SDA

Configuring head-end replication via DNAC

Multicast for your VNs can be configured from the DNAC itself and is fully automated. Traverse to your fabric page – Provision -> Fabric. Select your fabric and your site. You should see be able to left click on the site (next to the gears icon) and see an option to configure multicast like below:

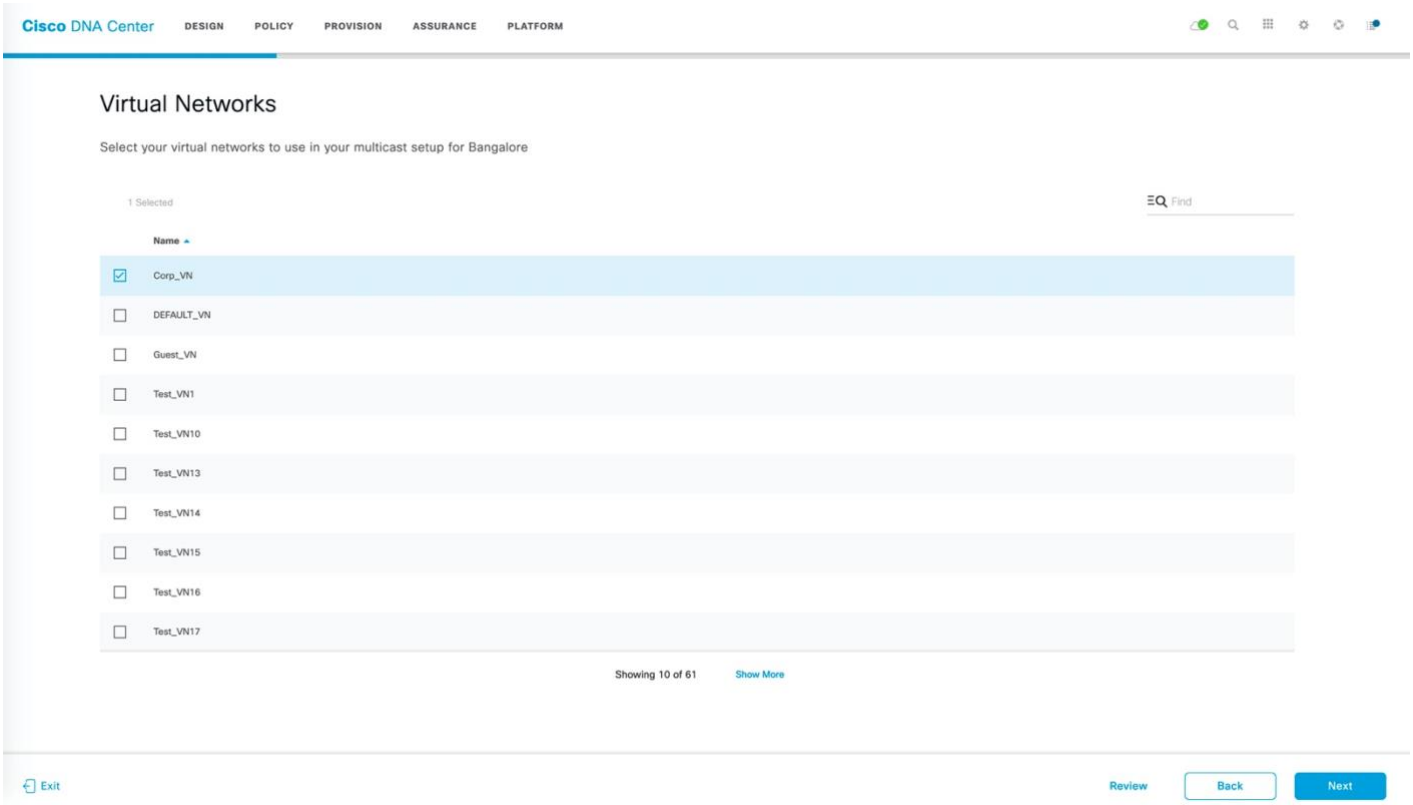


Choose “Head-end replication”:

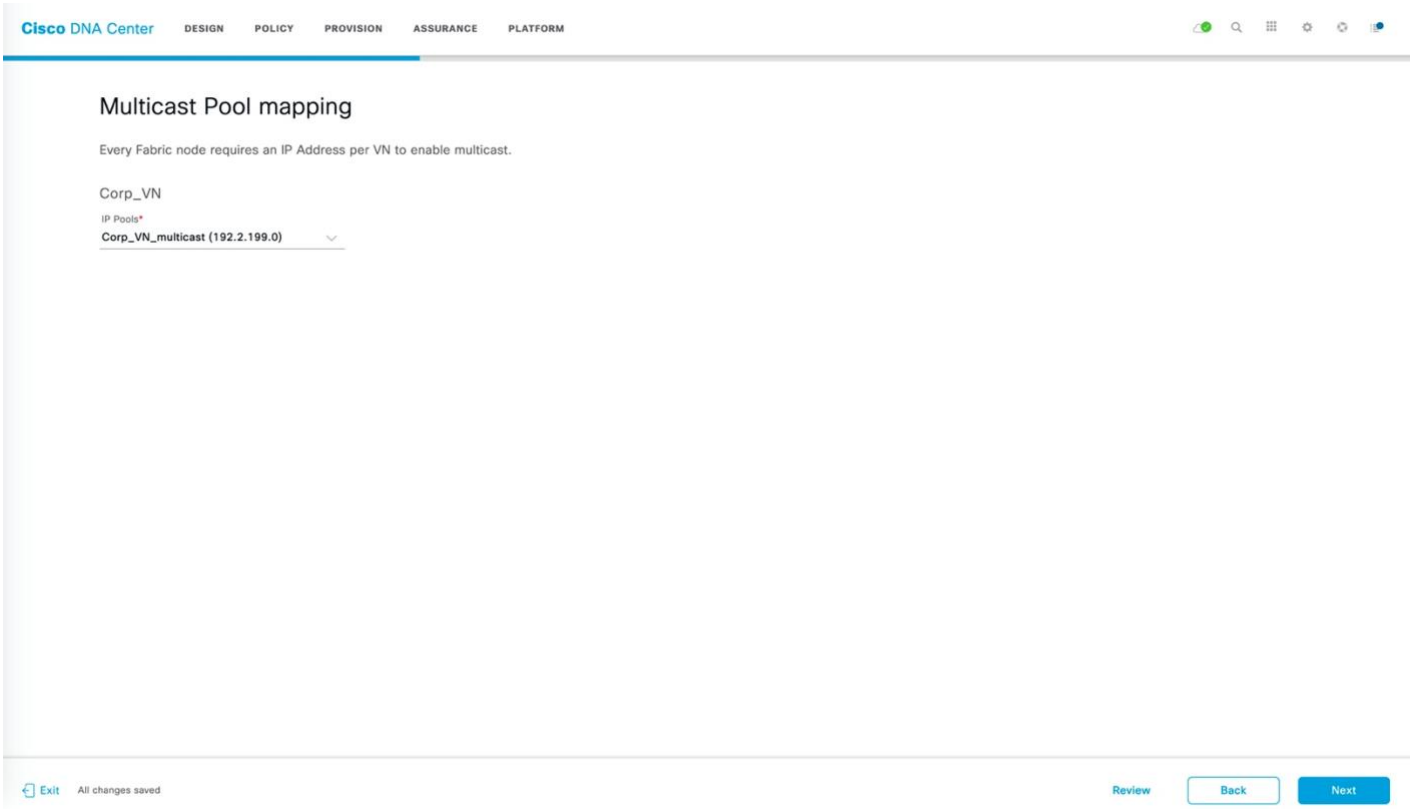


Aninda Chatterjee (aninchat@cisco.com)

Next, choose the VN you want to enable multicast for:



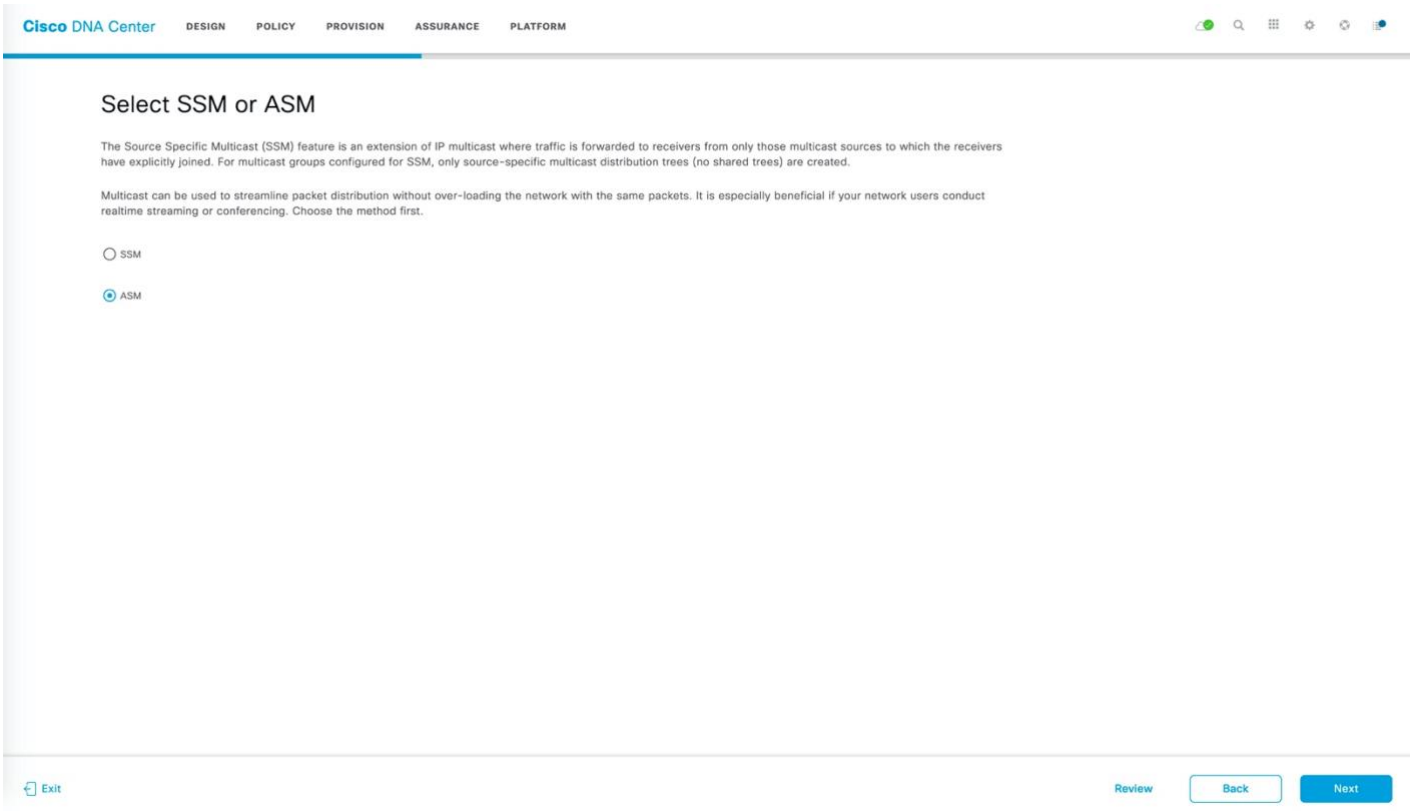
Next, choose a multicast pool:



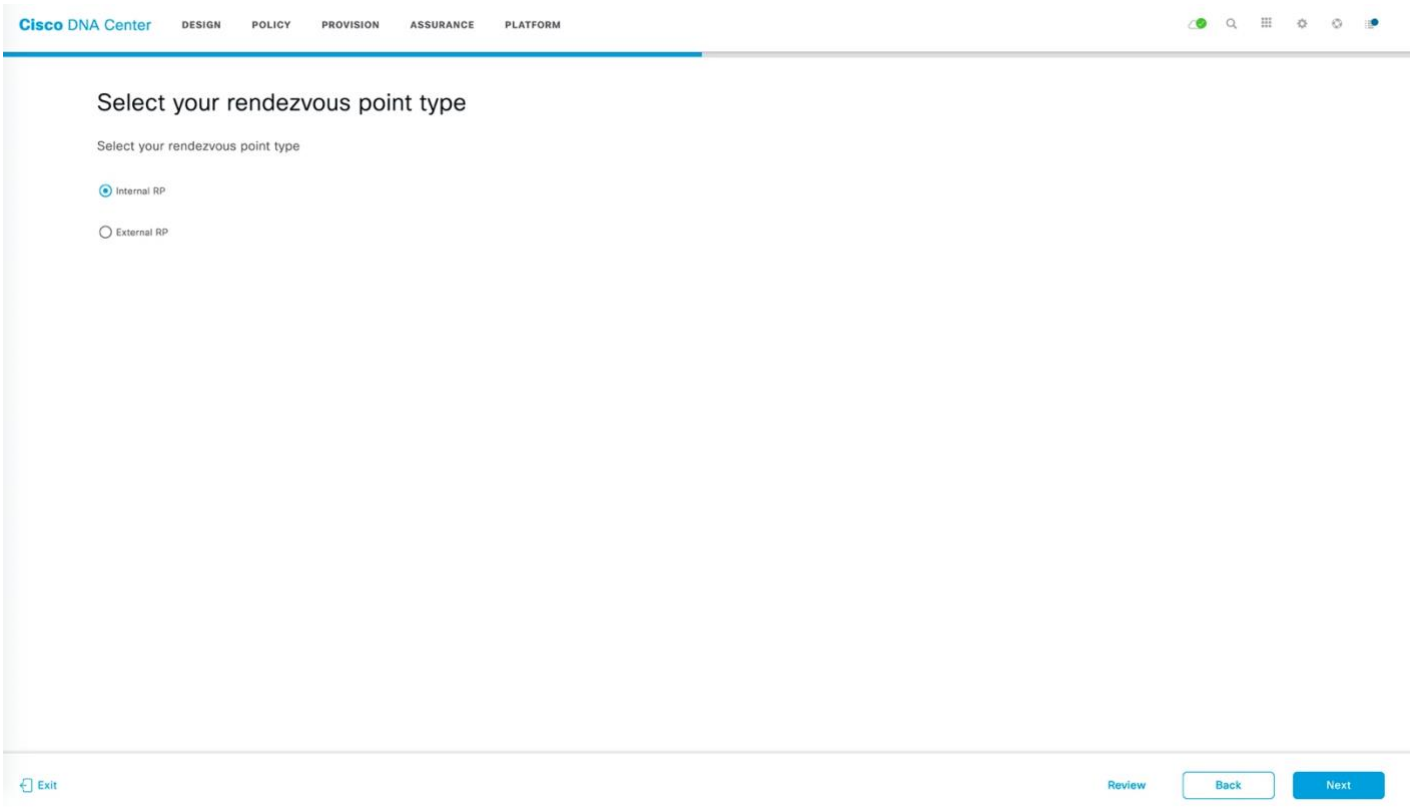
This pool must be pre-defined in DNAC and is used to create loopbacks on the fabric devices within the VRF.

Next, choose whether your overlay will run ASM or SSM (ASM will require a RP):

Aninda Chatterjee (aninchat@cisco.com)

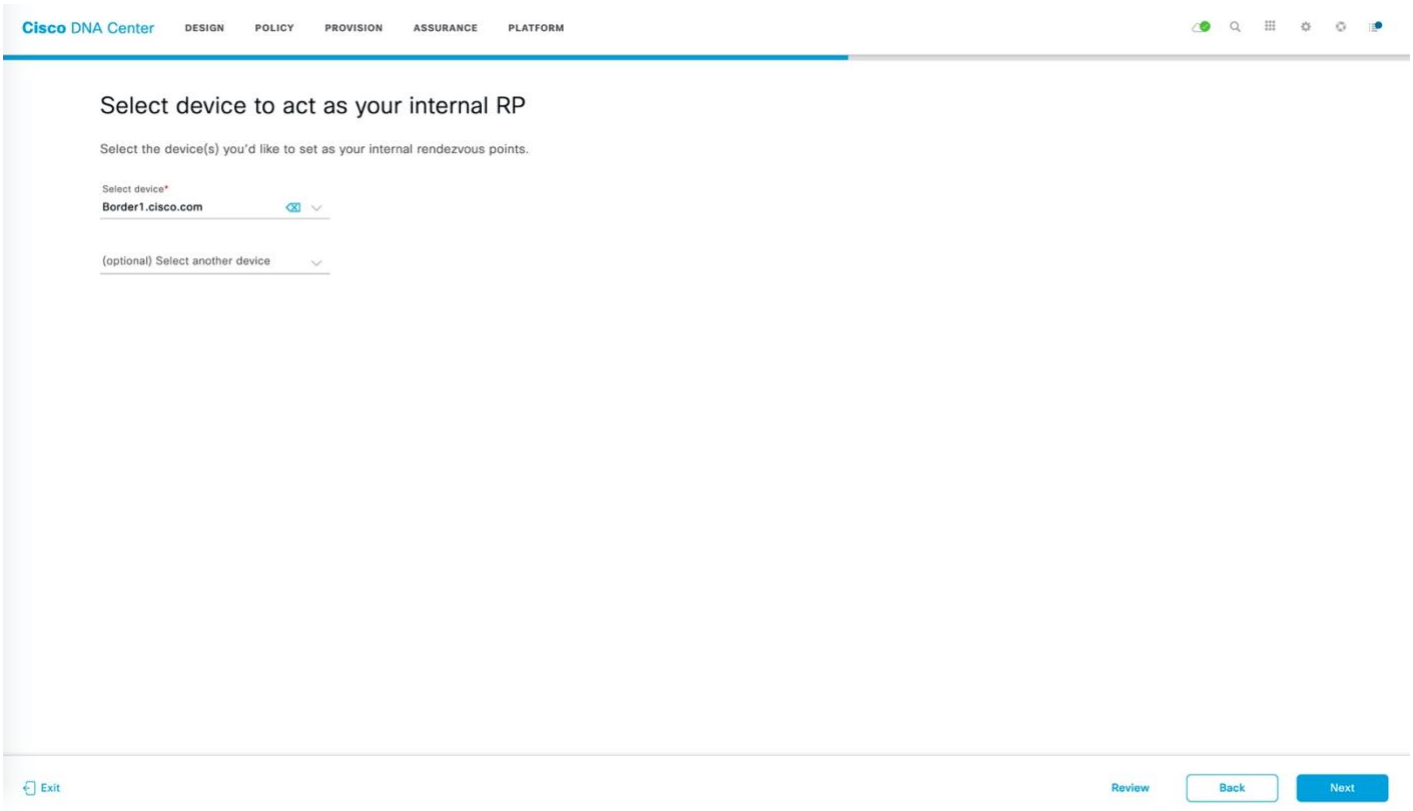


Next, choose where your RP is located – internal (inside the fabric) or external (outside the fabric). In our case, we want Border1 to be the RP, so we will choose internal.

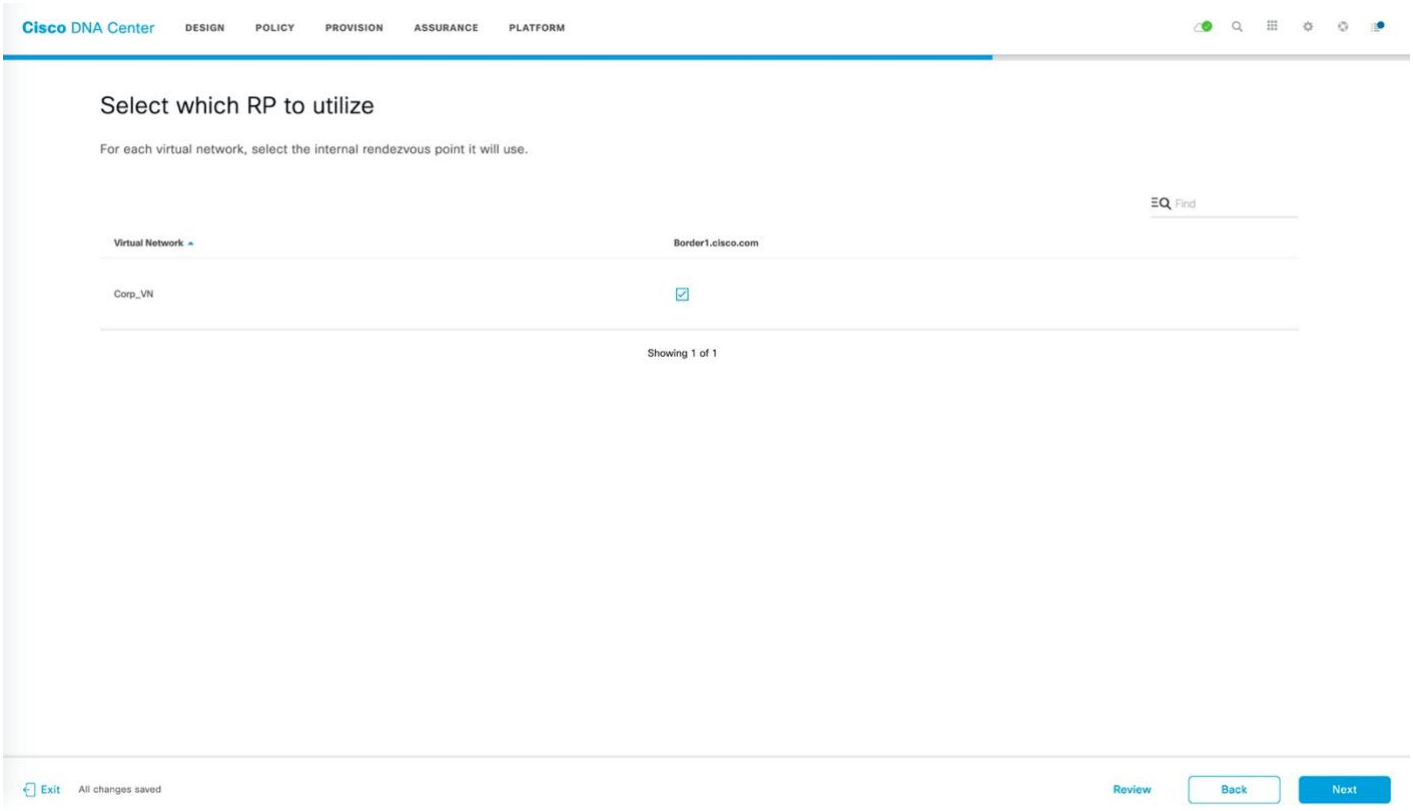


Next, select what device(s) should be the RP. We will select Border1 here.

Aninda Chatterjee (aninchat@cisco.com)

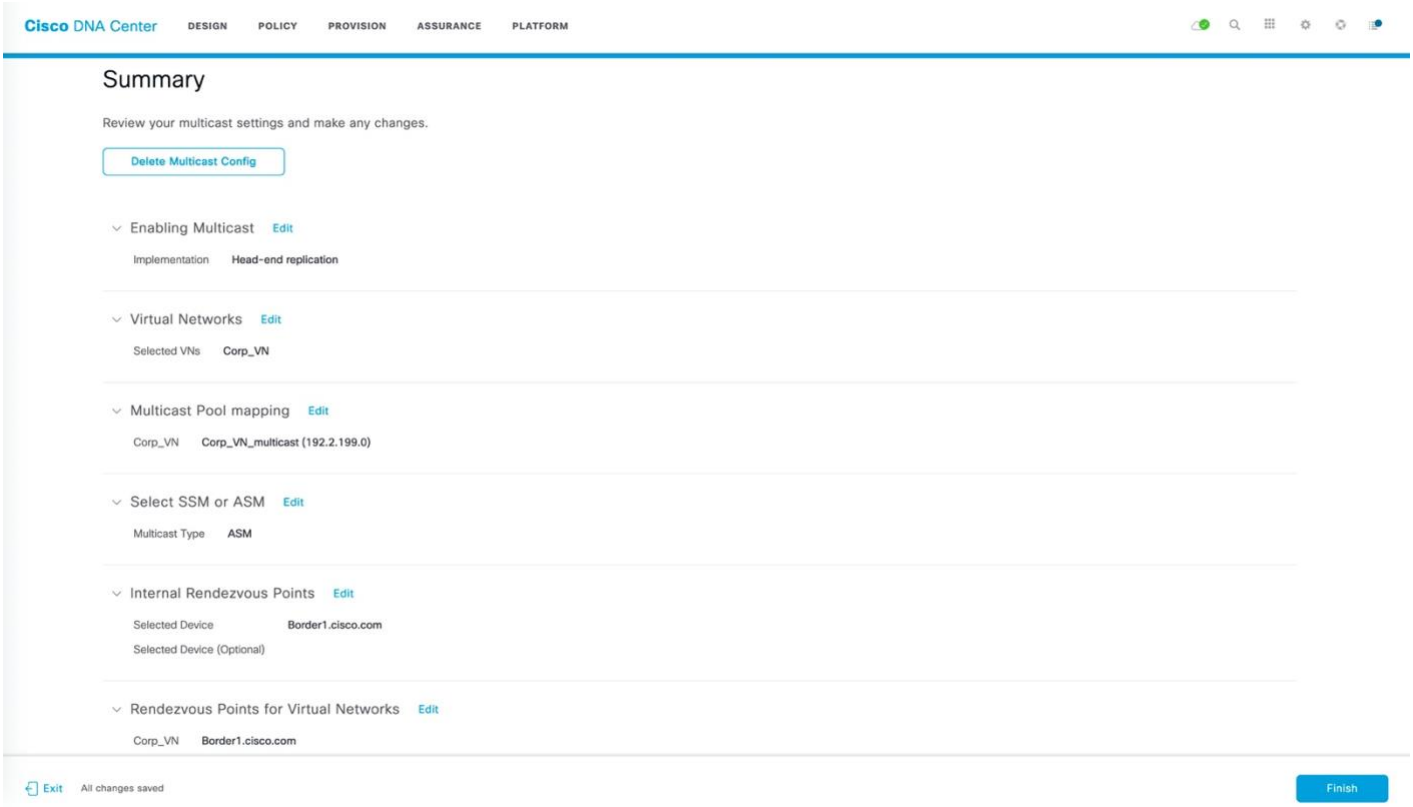


Next, select the RPs for your VNs.



Once you click ‘Next’, you should now see a summary of your choices:

Aninda Chatterjee (aninchat@cisco.com)



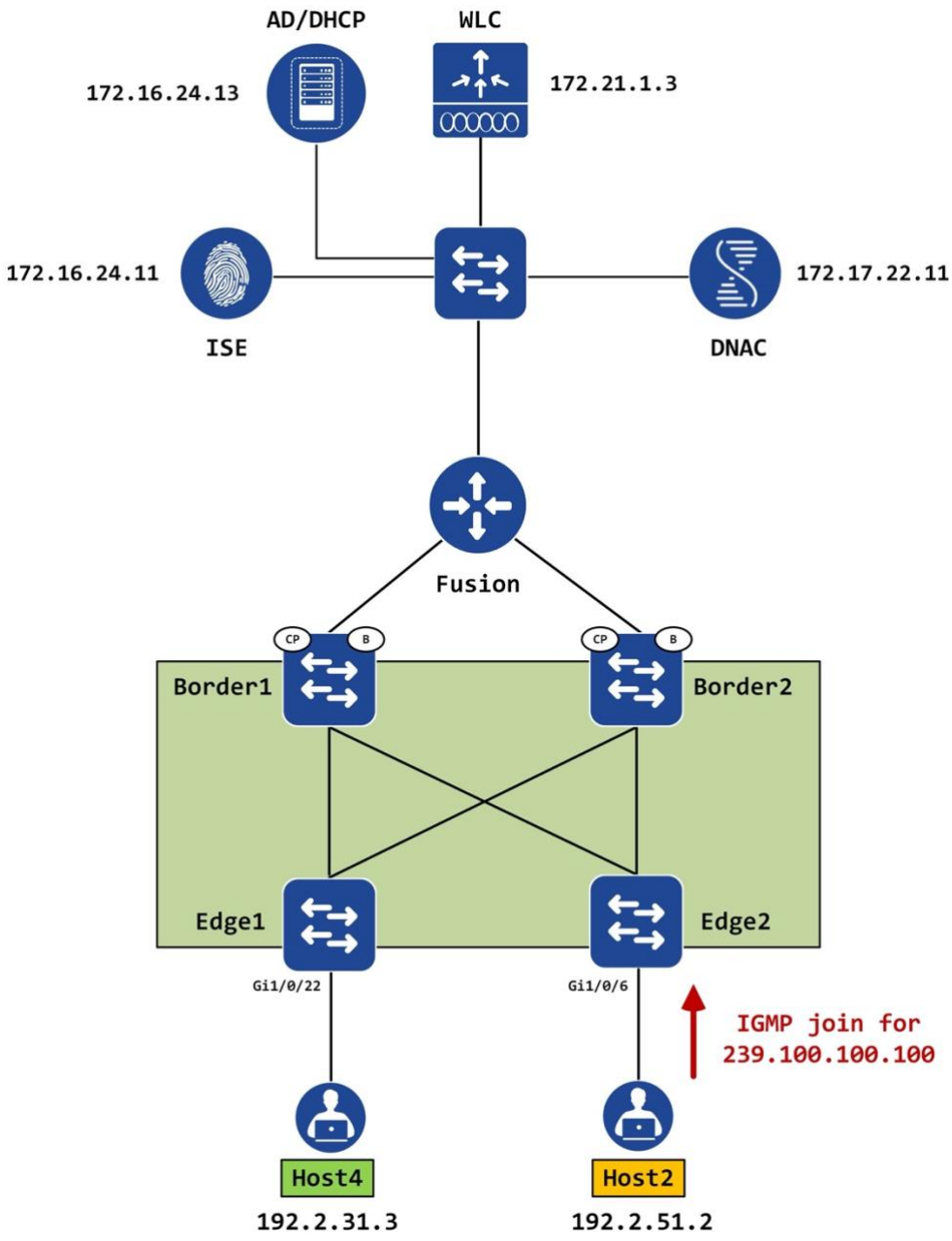
Once you click ‘Finish’, DNAC will build the appropriate configuration and push them to the devices.

Aninda Chatterjee (aninchat@cisco.com)

Source and Receiver in Fabric

Building the shared tree

Host2 sends an IGMPv3 join (without specifying a source, which essentially makes it an any source join for that group) for 239.100.100.100.




```
Edge2#show ip igmp vrf Corp_VN groups 239.100.100.100
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter   Group
Accounted
239.100.100.100  Vlan1030                 00:05:13  00:02:08  192.2.51.2
```


Aninda Chatterjee (aninchat@cisco.com)

```
Edge2#show ip igmp vrf Corp_VN membership 239.100.100.100
Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP, the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude

 Channel/Group                  Reporter        Uptime   Exp.  Flags  Interface
 *,239.100.100.100              0/1             00:05:23 01:58 3AT    Vl1030
```

A (*,G) entry is created in the VRF MRIB as well:

```
Edge2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:04:43/00:02:38, RP 192.2.199.1, flags: SJC
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 00:04:43/00:02:38
```

Several important things to dissect here. Remember that for ASM, it is essential to have a Rendezvous Point – both the shared and source trees are rooted at the RP. From the above, we can see that the RP is 192.2.199.1 – this is from configuration that was pushed during multicast deployment via DNAC.

The RP can also be confirmed using the following:

```
Edge2#show ip pim vrf Corp_VN rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 192.2.199.1 (?)
```


Aninda Chatterjee (aninchat@cisco.com)

The incoming interface is determined by a RPF (reverse path forwarding) lookup for the RP IP address. Let's confirm that it matches what we see in the output earlier:

```
Edge2#show ip cef vrf Corp_VN 192.2.199.1 detail
192.2.199.1/32, epoch 1, flags [subtree context, check lisp eligibility]
  SC owned,sourced: LISP remote EID - locator status bits 0x00000001
  LISP remote EID: 68392 packets 1367238 bytes fwd action encap, cfg as EID space
  LISP source path list
    nexthop 192.2.101.65 LISP0.4100
  2 IPL sources [no flags]
  nexthop 192.2.101.65 LISP0.4100
```

Good – so we know the RPF interface and next-hop is correctly populated in the multicast routing table and is in accordance with what we see in CEF. The outgoing interface in the multicast routing table is the SVI 1030 – this is where the host is connected and the L3 interface that processed the IGMP join.

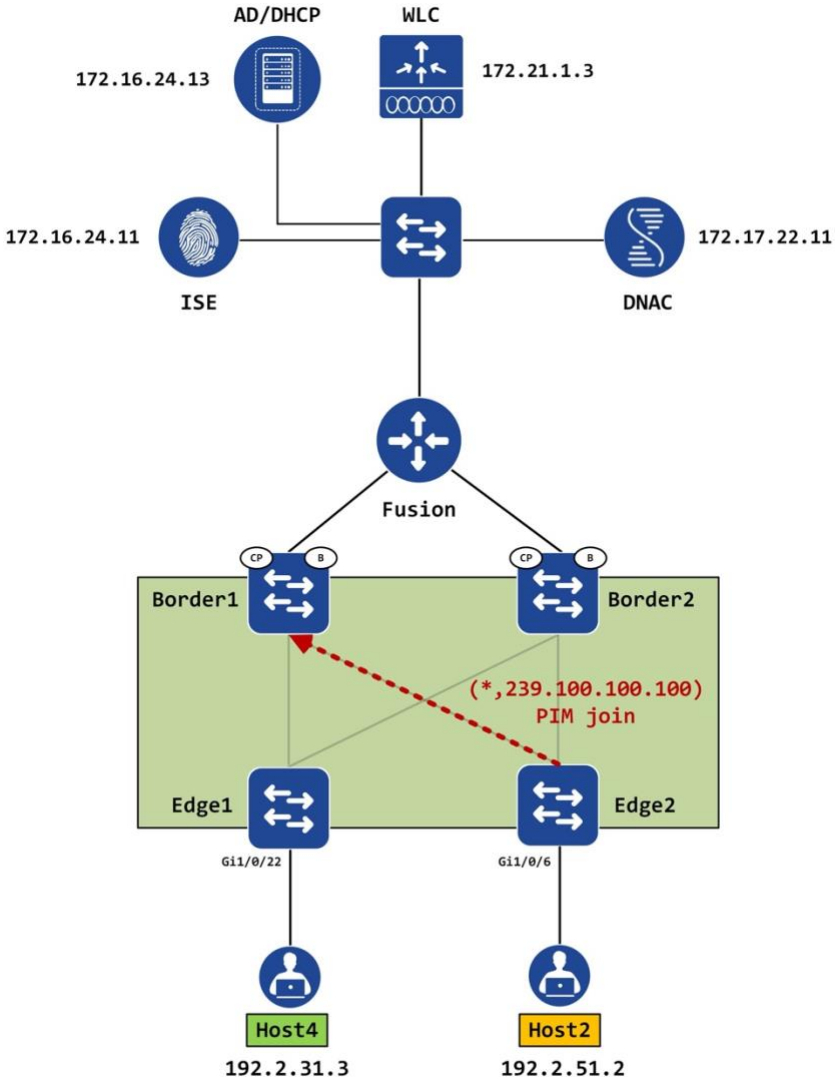
Remember, since this is headend replication, we do not run SSM in the underlay. How do you determine if you're using headend replication or native multicast? Simply look at the configuration under the LISP interface for this VRF.

```
Edge2#show run int lisp0.4100
Building configuration...

Current configuration : 48 bytes
!
interface LISP0.4100
 ip pim sparse-mode
end
```

By default, headend replication is used. To use underlay multicast as the replication transport, the CLI '*ip pim lisp transport multicast*' must be configured under the LISP interface.

Outside of the actual (*,G) entry creation on Edge2 itself, the IGMP join also triggers a PIM (*,G) join that is sent towards the RP (which is Border1) by Edge2.

Aninda Chatterjee (aninchat@cisco.com)



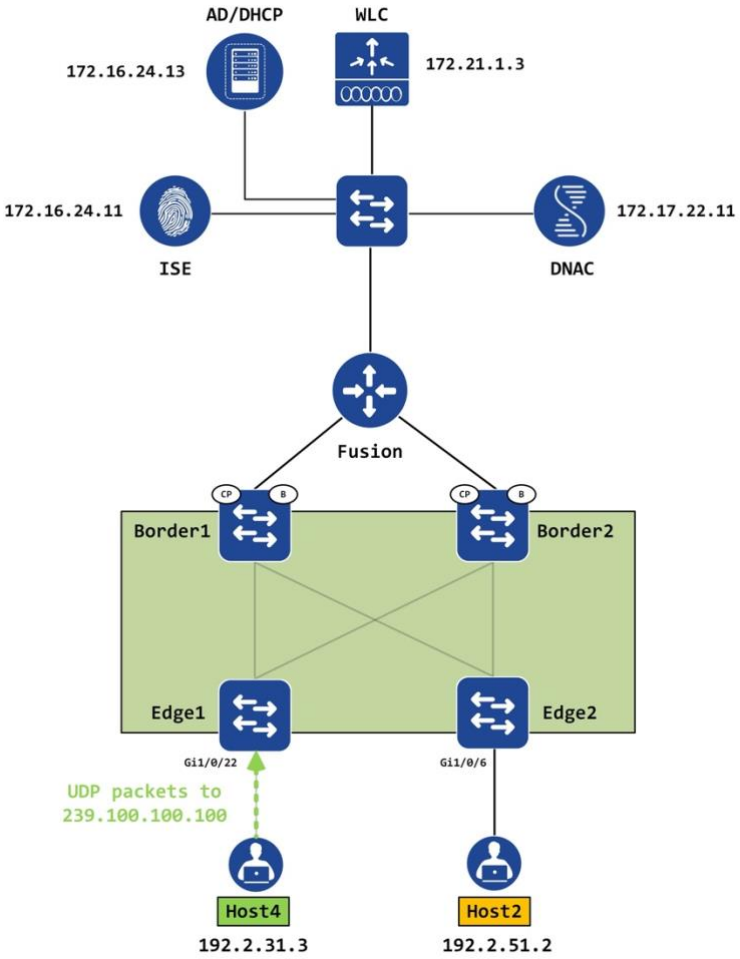
Border1 should now have a (*,G) entry for 239.100.100.100 in the overlay (in the corresponding VRF multicast routing table).

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 19:36:43/00:02:43, RP 192.2.199.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.71, Forward/Sparse, 19:36:43/00:02:43
```


Aninda Chatterjee (aninchat@cisco.com)

Start of source stream

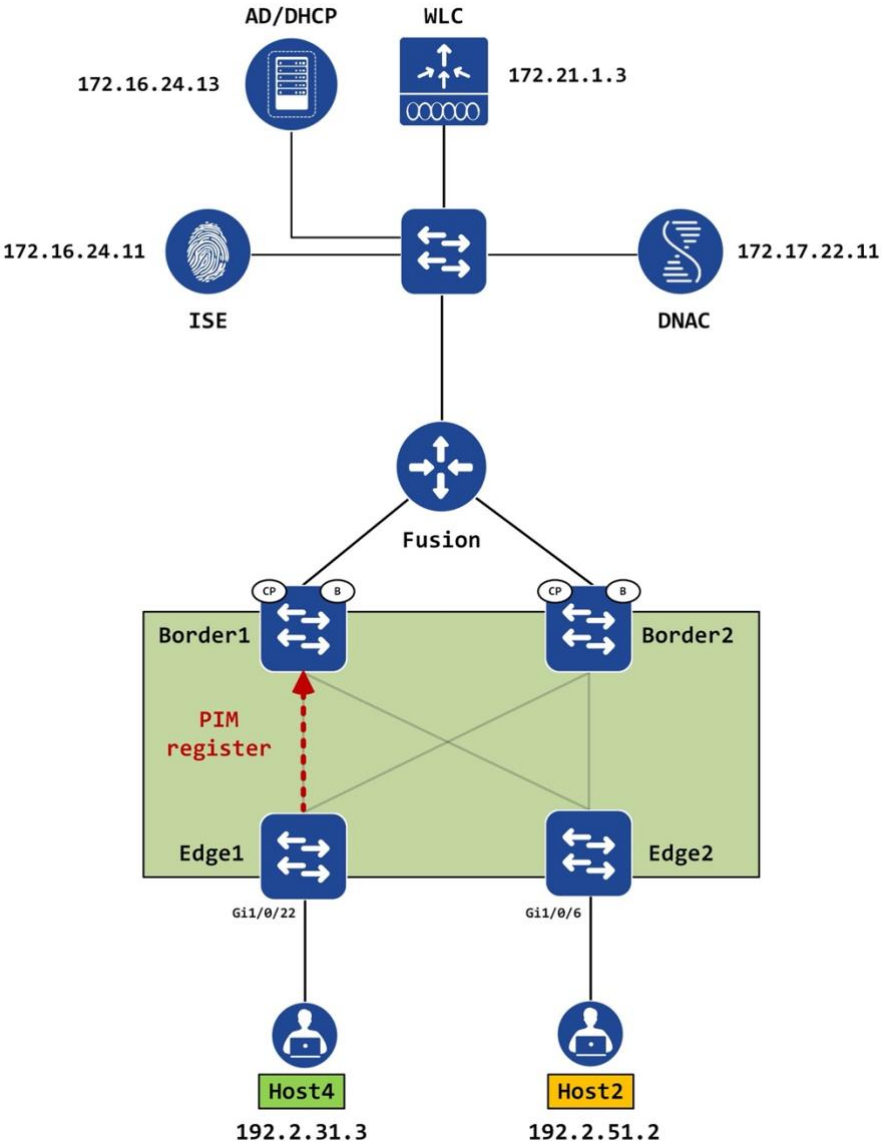
The source, Host4, now starts to send UDP traffic to 239.100.100.100. When this hits Edge1, because of a miss in MFIB, the packet is punted to the CPU to create state in the multicast routing table.




```
// no entry found in mfib, packet is punted to form state

014401: Jul 15 17:57:08.104: FMANRP-MCAST: (192.2.31.3, 239.100.100.100) mfib entry not
found
014402: Jul 15 17:57:08.104: FMANRP-MCAST: (*, 239.100.100.100) mfib entry not found
014403: Jul 15 17:57:08.104: FMANRP-MCAST: Punted packet from Vlan1029 tagged.
014404: Jul 15 17:57:08.124: FMANRP-MCAST(2): add (192.2.31.3, 239.100.100.100/64), id
0x0

// mrib adds entry

014416: Jul 15 17:57:08.125: MRIB: Table = VRF Corp_VN, [4  ] i/o - received 36 bytes
014417: Jul 15 17:57:08.125: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Received   DDE* [Vl1029  SP* ]
014418: Jul 15 17:57:08.125: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Updated   DDE* #A=0, #F=0 [Vl1029  SP* ]
014419: Jul 15 17:57:08.125: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Added
```


Aninda Chatterjee (aninchat@cisco.com)

'DDE' in the above debugs stands for 'Data Driven Event', which essentially means the state creation was due to the data packet that was punted to the CPU. Since Edge1 is the first hop router, it must register this source to the RP (Border1). It does so by sending via unicast, encapsulating the actual multicast data and sending it to the RP. During this time, the 'Registering' flag is set as well.



The following debugs on Edge1 show this as well:

```
014473: Jul 15 17:57:08.128: PIM(2): Adding register encap tunnel (Tunnel2) as
forwarding interface of (192.2.31.3, 239.100.100.100).
014474: Jul 15 17:57:08.128: MRT(2): Set the F-flag for (*, 239.100.100.100)
014475: Jul 15 17:57:08.128: MRT(2): Set the F-flag for (192.2.31.3, 239.100.100.100)
014476: Jul 15 17:57:08.128: MRT(2): Start 0x7F78F9BD9818 [2778] send join timer
(192.2.31.3, 239.100.100.100) delay 58 sec , rpf_nbr 0.0.0.0/0.0.0.0
014477: Jul 15 17:57:08.128: MRT(2): Create (192.2.31.3,239.100.100.100), RPF
(Vlan1029, 0.0.0.0, 10/1)
014483: Jul 15 17:57:08.129: MRT(2): Set the T-flag for (192.2.31.3, 239.100.100.100)
```

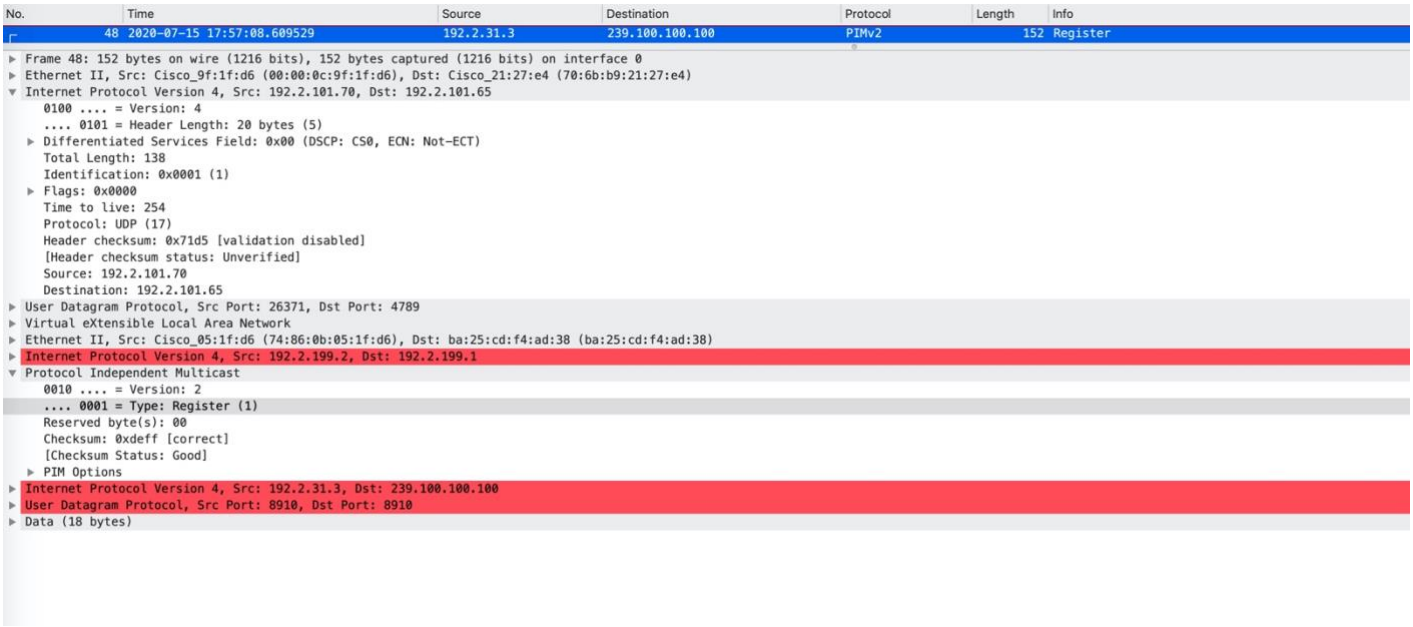
The MRIB output during the register process, with the register flag set:

Aninda Chatterjee (aninchat@cisco.com)

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:00:00/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65, LISP: [192.2.101.65, 232.0.3.1]
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 00:00:00/00:02:59, flags: FTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.1), Forward/Sparse, 00:00:00/stopped, p
      192.2.101.65, 00:00:00/00:03:29
```

Notice that the outgoing interface list contains Border1s IP address. The incoming interface points to VLAN1029, which is where the source is connected.

The register packet looks like this:



Aninda Chatterjee (aninchat@cisco.com)

The inner most packet is the actual multicast data here. A PIM header is added on top of it with a type of “Register”. This is then encapsulated into a unicast IPv4 header, destined to the RP – 192.2.199.1. Remember, this now has to traverse the fabric and the RP is known via LISP in the overlay.

```
Border1#show lisp instance-id 4100 ipv4 server 192.2.199.1
LISP Site Registration Information

Site name: site_uci
Description: map-server configured from Cisco DNA-Center
Allowed configured locators: any
Requested EID-prefix:

  EID-prefix: 192.2.199.1/32 instance-id 4100
    First registered:     20:56:30
    Last registered:      20:56:22
    Routing table tag:    0
    Origin:               Dynamic, more specific of 192.2.199.0/24
    Merge active:         Yes
    Proxy reply:          Yes
    TTL:                  1d00h
    State:                complete
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 192.2.101.65:32927, last registered 20:56:22, proxy-reply, map-notify
                            TTL 1d00h, merge, hash-function sha1, nonce 0x164A4EFC-
0xCDFE63F6
                            state complete, no security-capability
                            xTR-ID 0x5A97B19C-0xCEBF6332-0x4EE1E4F1-0xFD7CFA67
                            site-ID unspecified
                            sourced by reliable transport
      Locator       Local  State      Pri/Wgt  Scope
      192.2.101.65  yes    up          10/10   IPv4 none
    Merged locators
      Locator       Local  State      Pri/Wgt  Scope        Registering ETR
      192.2.101.65  yes    up          10/10   IPv4 none    192.2.101.65:32927
```

On Edge1, we can see that the CEF entry points to LISP:

```
Edge1#show ip cef vrf Corp_VN 192.2.199.1 detail
192.2.199.1/32, epoch 1, flags [subtree context, check lisp eligibility]
  SC owned,sourced: LISP remote EID - locator status bits 0x00000001
  LISP remote EID: 9097 packets 734486 bytes fwd action encap, cfg as EID space
  LISP source path list
    nexthop 192.2.101.65 LISP0.4100
  2 IPL sources [no flags]
  nexthop 192.2.101.65 LISP0.4100
```

Thus, the final encapsulation is VXLAN, with the outer IPv4 header destined to 192.2.101.65.

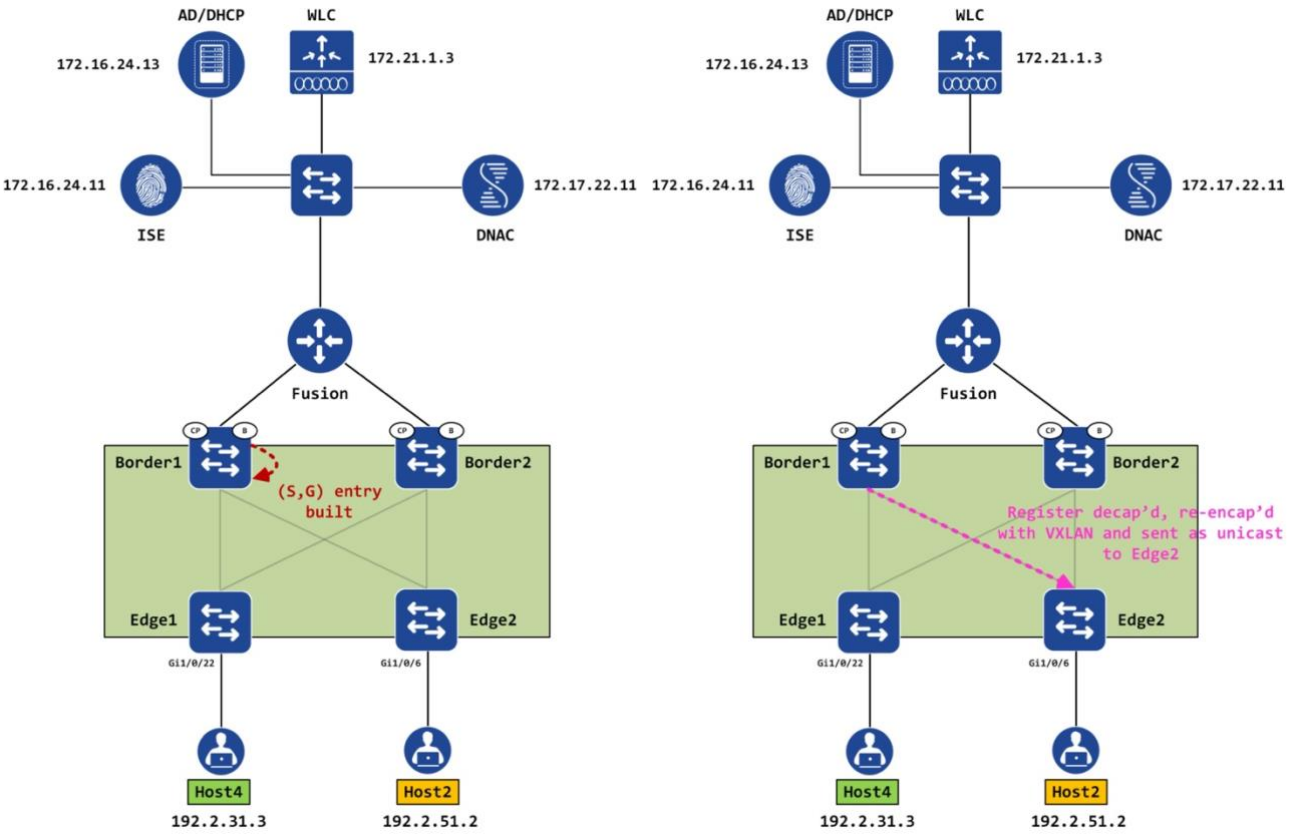
The register packet does two things on the RP (Border1): first, the packet is decapsulated and sent towards the receiver using the shared tree that is already built.

Aninda Chatterjee (aninchat@cisco.com)

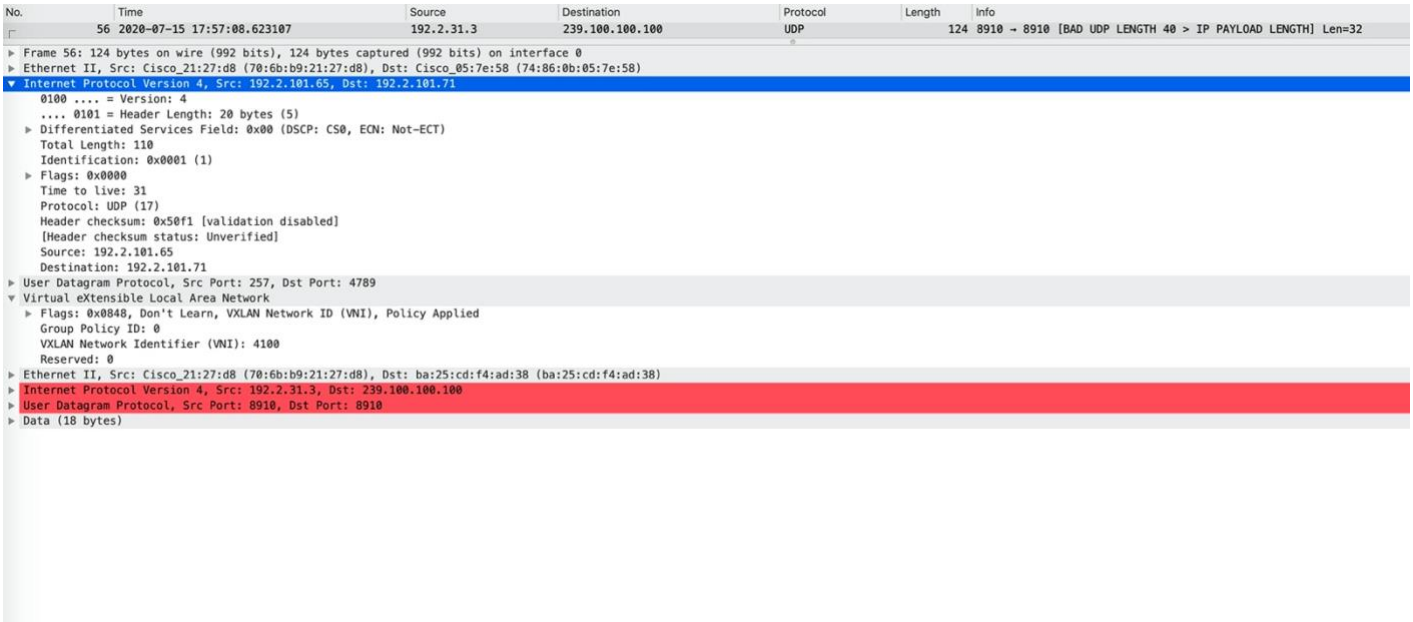
Second, it also builds (S,G) state on the RP (Border1) which leads to (S,G) PIM joins sent towards the source S, hop by hop.

```
// Border1 receives register

Jul 15 17:57:08.610: PIM(117): Received v2 Register on LISP0.4100 from 192.2.199.2
Jul 15 17:57:08.610:      for 192.2.31.3, group 239.100.100.100

// (S,G) state built from register

Jul 15 17:57:08.614: PIM(117): Adding register decap tunnel (Tunnel3) as accepting
interface of (192.2.31.3, 239.100.100.100).
Jul 15 17:57:08.616: MRT(117): Create (192.2.31.3,239.100.100.100), RPF (unknown,
0.0.0.0, 0/0)
Jul 15 17:57:08.618: MRT(117): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 192.2.101.71 Outer-source: 0.0.0.0 Successful
Jul 15 17:57:08.621: MRT(117): Add LISP0.4100/192.2.101.71 to the olist of (192.2.31.3,
239.100.100.100), Forward state - MAC not built
```

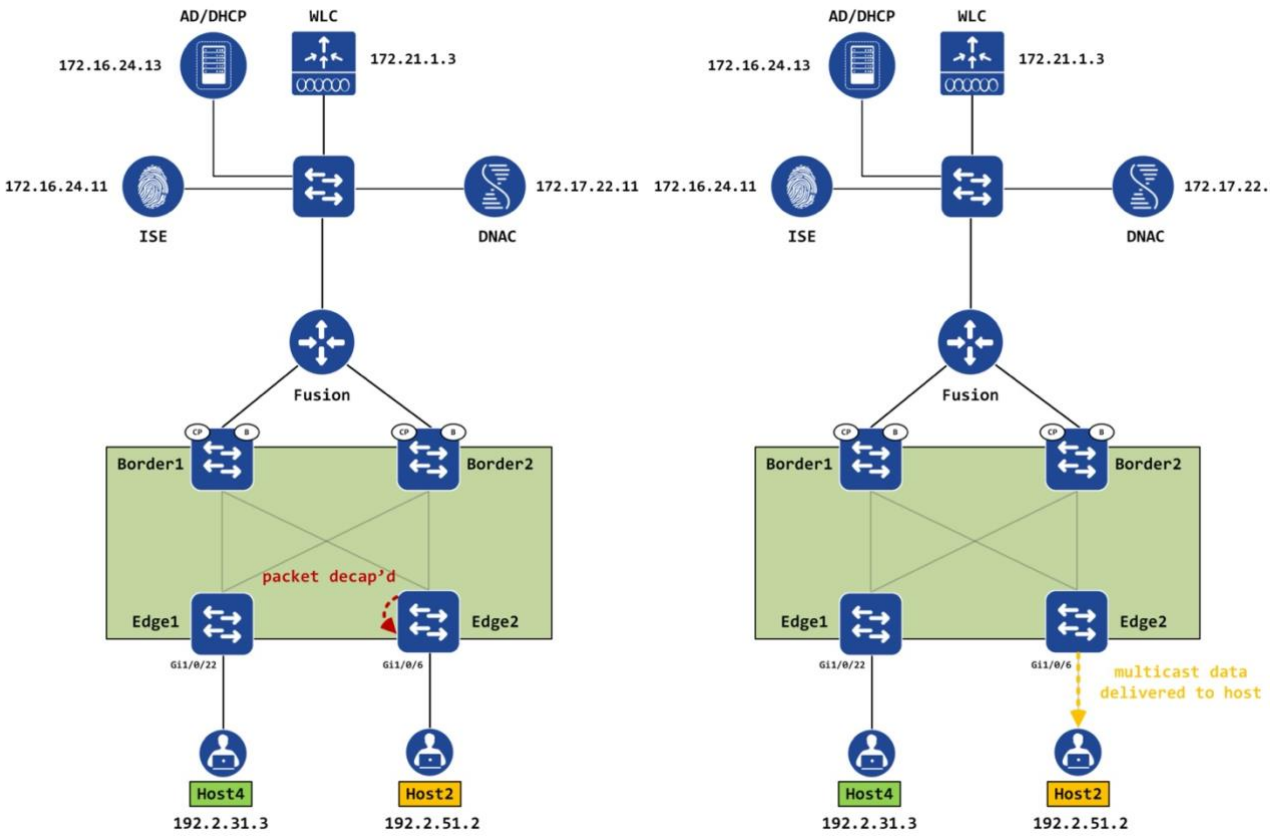
The decapsulation of register and forwarding of the underlying multicast packet can be visualized like so:



This packet looks like this:



As you can see, the outer header is now a unicast IPv4 header, with a source of Border1 and destined to Edge2. Edge2 will simply decapsulate the packet and forward the inner multicast data to the receiver:

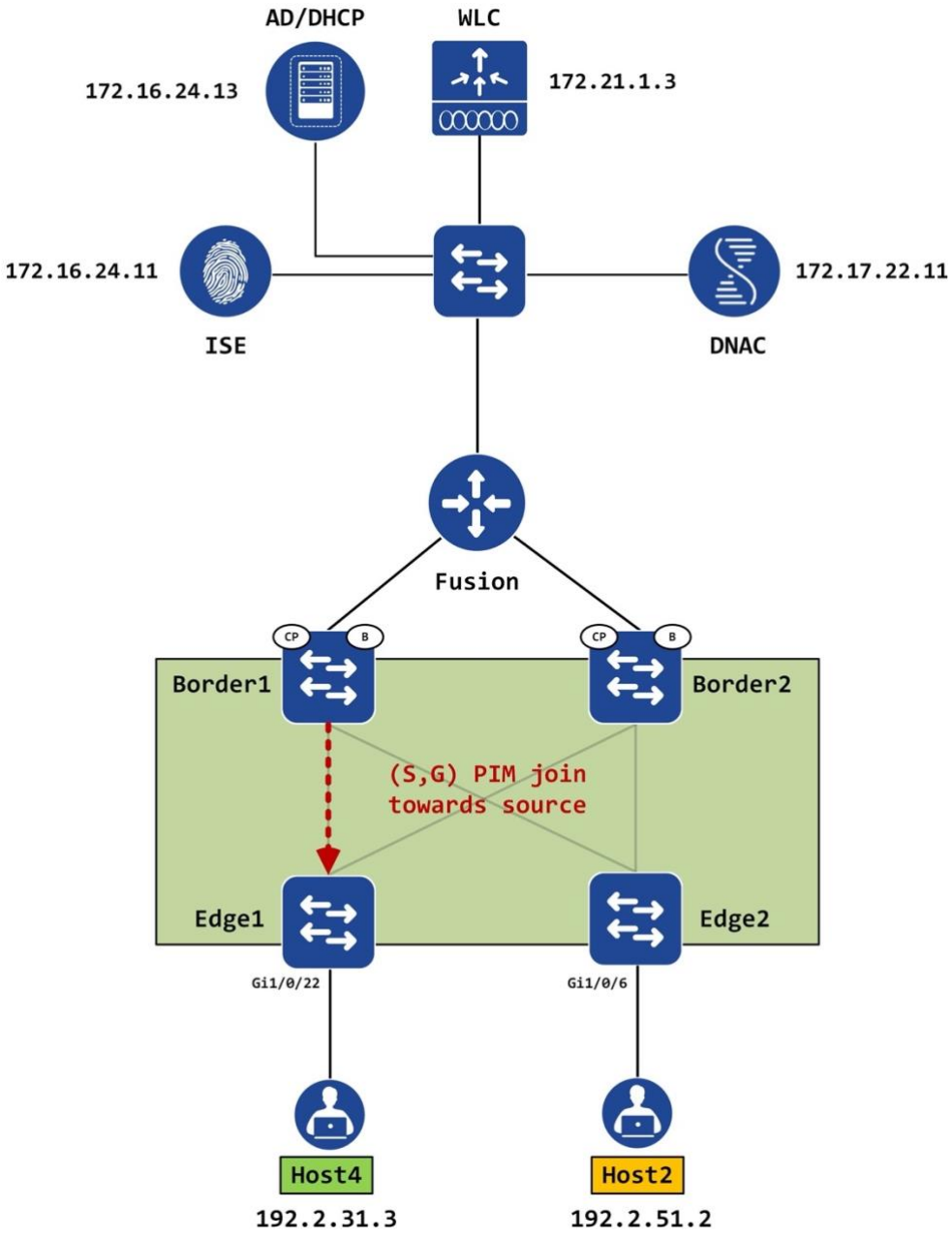


This process ensures that while the source is registering to the RP, we don't lose any multicast data during this period of time.

Building the source tree

Once the (S,G) state is built on the RP (Border1), it will send a PIM (S,G) join towards the source S. In our case, a (192.2.31.3, 239.100.100.100) PIM join should be sent towards 192.2.31.3.

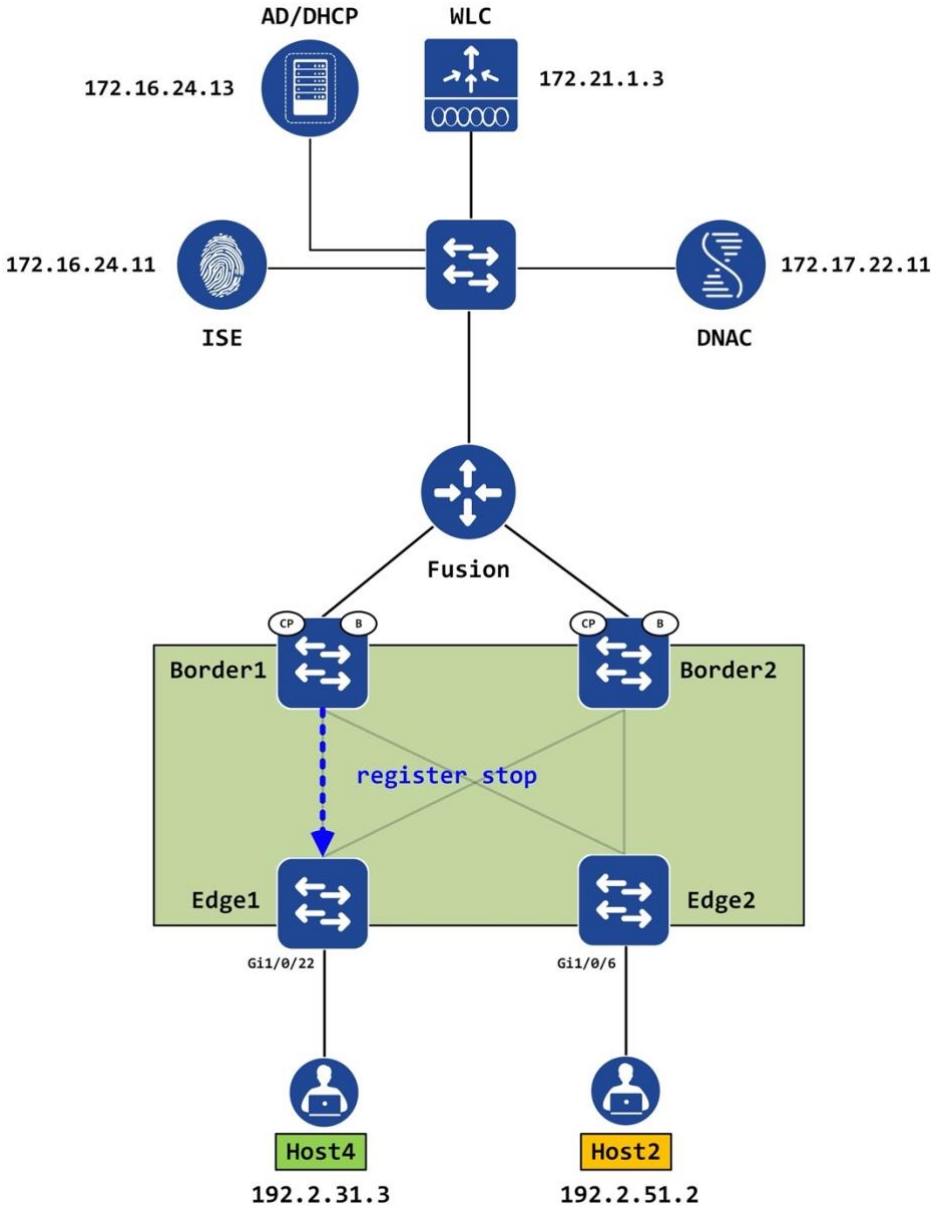
```
// (S,G) PIM join sent to Edge1

Jul 16 18:26:17.178: PIM(117): Insert (192.2.31.3,239.100.100.100) join in nbr
192.2.101.70's queue
Jul 16 18:26:17.178: PIM(117): Building Join/Prune packet for nbr 192.2.101.70
Jul 16 18:26:17.178: PIM(117):  Adding v2 (192.2.31.3/32, 239.100.100.100), S-bit Join
Jul 16 18:26:17.178: PIM(117): Adding LISP Unicast transport attribute in join/prune to
192.2.101.70 (LISP0.4100)
Jul 16 18:26:17.179: PIM(117): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```



This join is received by Edge1 and the existing (S,G) entry is updated and the underlay SSM entry is correctly built as well.

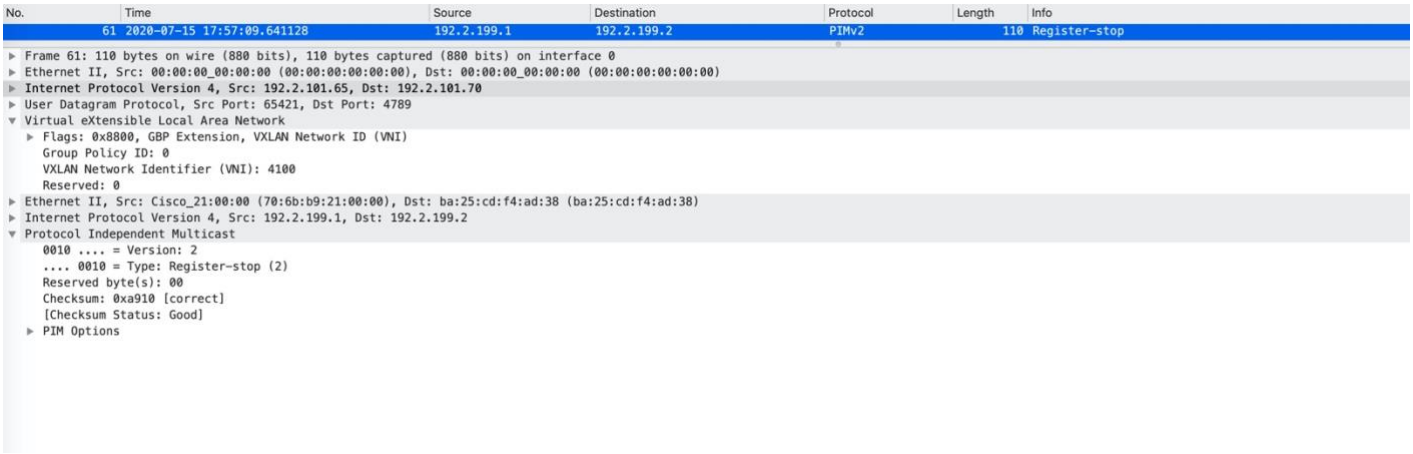
```
// (S,G) state updated on Edge1 after receiving PIM join from Border1

018284: Jul 16 18:26:17.181: PIM(2): J/P Transport Attribute, Transport Type: Unicast
018285: Jul 16 18:26:17.181: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
018286: Jul 16 18:26:17.181: MRT(2): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 192.2.101.65 Outer-source: 0.0.0.0 Successful
018294: Jul 16 18:26:17.181: MRT(2): Add LISP0.4100/192.2.101.65 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
018295: Jul 16 18:26:17.181: PIM(2): Add LISP0.4100/192.2.101.65 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join
```

Once the (S,G) is built on Edge1, it starts to natively forward the multicast stream as well (along with the registers). On receiving multicast natively, the RP knows that the source tree has been built and it can now send a register stop to Edge1.



Aninda Chatterjee (aninchat@cisco.com)

The register stop packet:



Once Edge1 receives the register stop, it can now stop sending register messages to the RP, clear the register flag and only forward the multicast stream natively.

```
// Edge1 receives register stop

018348: Jul 16 18:26:17.646: PIM(2): Received v2 Register-Stop on LISP0.4100 from
192.2.199.1
018349: Jul 16 18:26:17.646: PIM(2):   for source 192.2.31.3, group 239.100.100.100
018352: Jul 16 18:26:17.646: PIM(2): Removing register encap tunnel (Tunnel2) as
forwarding interface of (192.2.31.3, 239.100.100.100).
018354: Jul 16 18:26:17.646: PIM(2): Clear Registering flag to 192.2.199.1 for
(192.2.31.3/32, 239.100.100.100)
```

SPT cutover

At the same time, the register packet that was decapsulated and sent towards the receiver over the shared tree creates a (S,G) state in its MRIB after a MFIB miss. This is where things get really interesting.

```
// MFIB miss creates (S,G) state

139714: Jul 16 18:26:17.140: FMANRP-MCAST: (192.2.31.3, 239.100.100.100) mfib entry not
found
139715: Jul 16 18:26:17.140: FMANRP-MCAST: (*, 239.100.100.100) mfib entry not found
139716: Jul 16 18:26:17.159: FMANRP-MCAST(2): add (192.2.31.3, 239.100.100.100/64), id
0x0
```

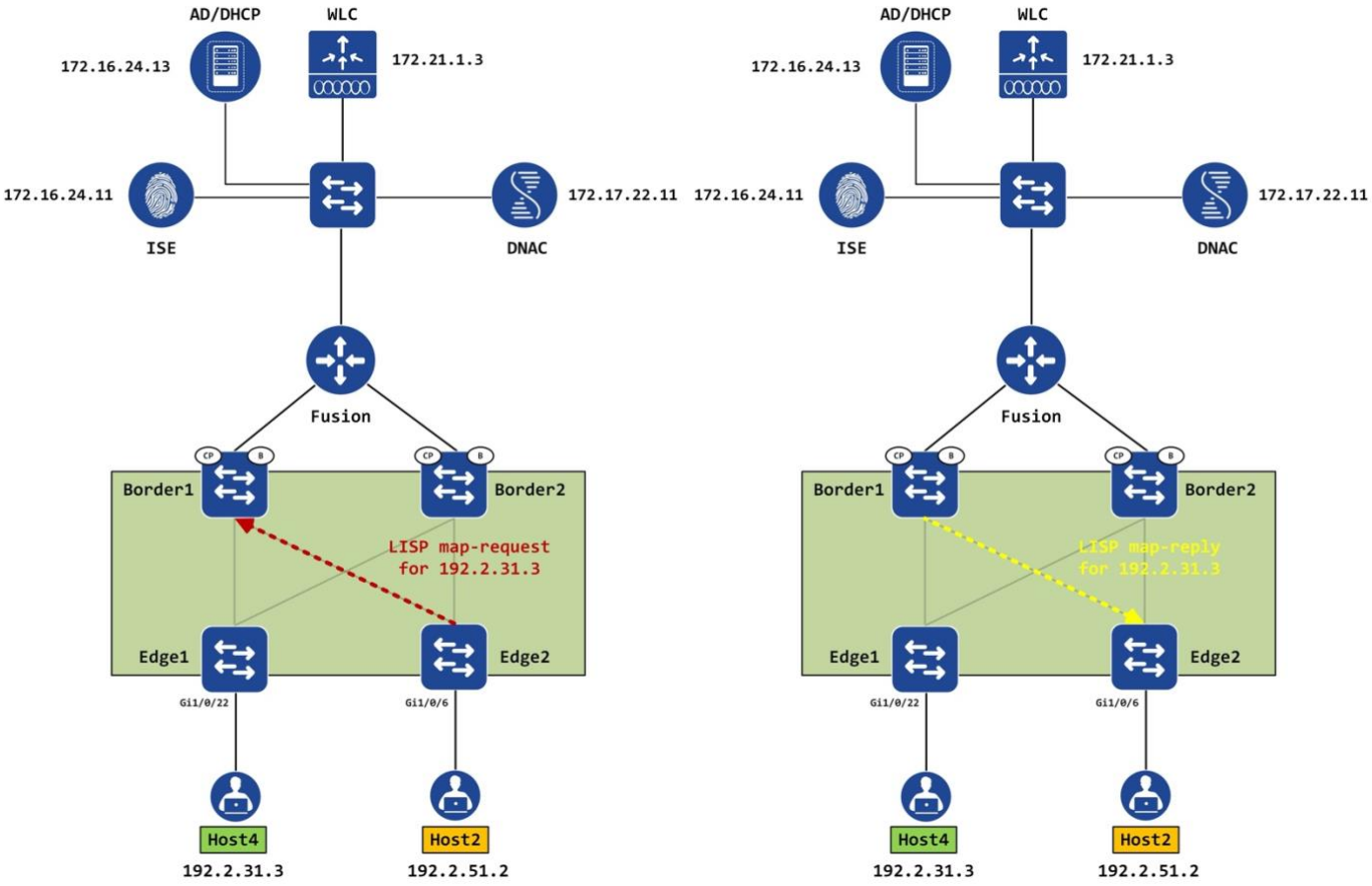
```
// (S,G) state created in multicast routing table

139728: Jul 16 18:26:17.160: MRIB: Table = VRF Corp_VN, [4  ] i/o - received 36 bytes
139729: Jul 16 18:26:17.160: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Received   DDE* [LI0.4100  SP* ]
139730: Jul 16 18:26:17.160: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Updated   DDE* #A=0, #F=0 [LI0.4100  SP* ]
139731: Jul 16 18:26:17.161: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Added
139775: Jul 16 18:26:17.164: MRT(2): Create (192.2.31.3,239.100.100.100), RPF
(Vlan1029, 0.0.0.0, 0/0)
139781: Jul 16 18:26:17.164: MRT(2): WAVL Insert interface: Vlan1030 in
(192.2.31.3,239.100.100.100) Successful
139794: Jul 16 18:26:17.165: MRT(2): Add Vlan1030/239.100.100.100 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
139795: Jul 16 18:26:17.165: MRT(2): Set the J-flag for (192.2.31.3, 239.100.100.100)
```

After this, a process called route watch is triggered for the source in order to determine where the source is located.

```
// rwatch triggered for source, along with RPF track for the source

139746: Jul 16 18:26:17.162: MCAST-RWATCH: -af IPv4 -topo 8002: Route watch topology
found.
139747: Jul 16 18:26:17.162: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix not found.
139748: Jul 16 18:26:17.162: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix created.
139749: Jul 16 18:26:17.162: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in RIB.
139750: Jul 16 18:26:17.162: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in LISP.
139751: Jul 16 18:26:17.163: MRT(2): RPF Track start on 192.2.31.3 for (192.2.31.3,
239.100.100.100)
```

This route watch process triggers a resolution of the source via LISP. This leads to the entire process of sending out a map-request for 192.2.31.3, processing a map-reply, updating the LISP map-cache and eventually pushing this information in RIB/FIB.

Aninda Chatterjee (aninchat@cisco.com)



Once the LISP resolution is complete, an update is sent to RIB and to route watch informing of the event completion.

```
// send update back to rwatch and mark prefix as LISP learnt

139876: Jul 16 18:26:17.171: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Received LISP notification, masklen = 32.
139877: Jul 16 18:26:17.171: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Prefix status set to: LISP.
139878: Jul 16 18:26:17.171: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Prefix enqueued for notification.
```

At the end of this process, Edge2 knows exactly where the source is located and what the RLOC IP address. Finally, the RPF interface for the (S,G) entry is updated to point to the source RLOC for 192.2.31.3, which is 192.2.101.70.

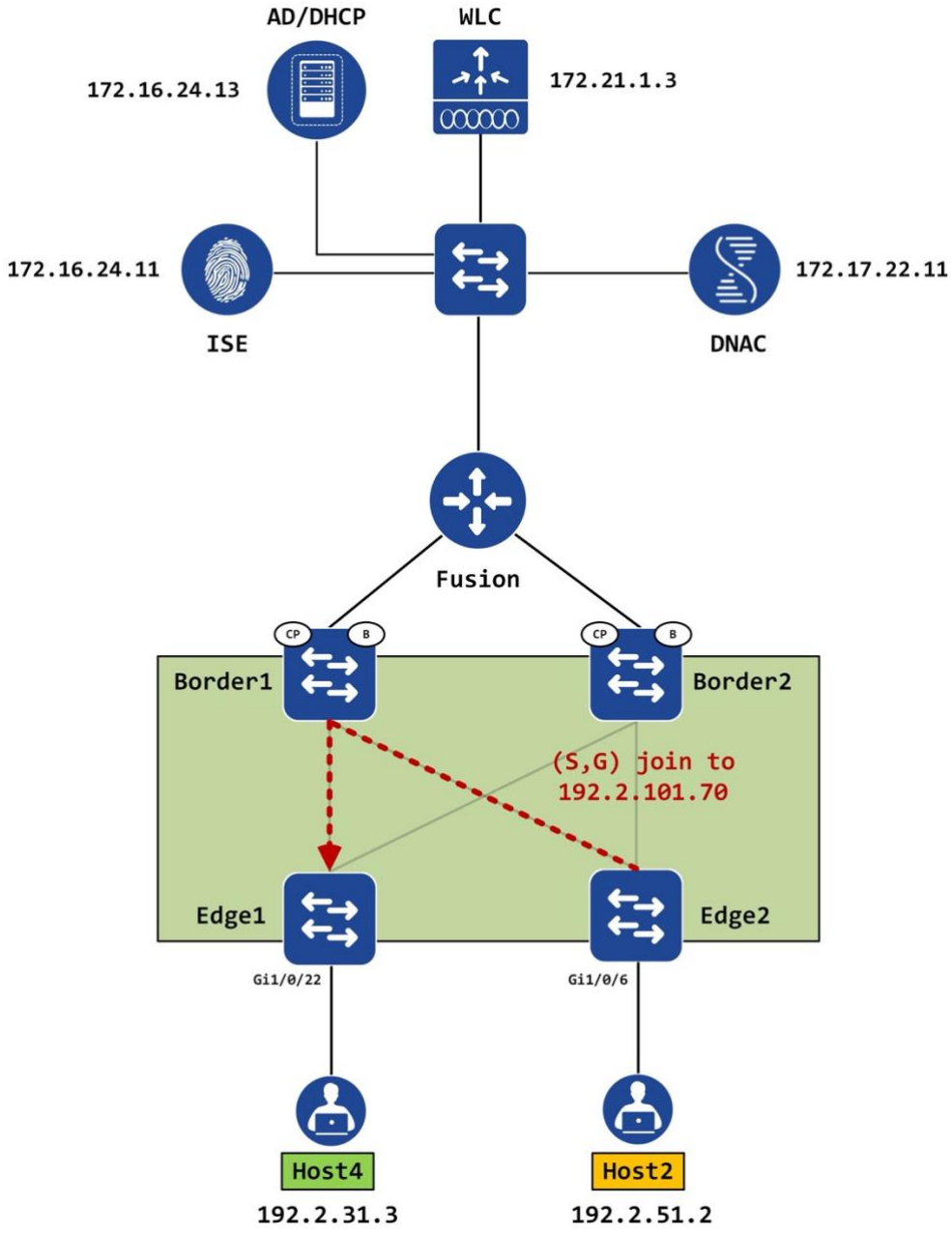
```
// update RPF on (S,G) entry to point to LISP interface/RLOC IP

139882: Jul 16 18:26:17.172: MRT(2): (192.2.31.3,239.100.100.100), RPF change from
Vlan1029/0.0.0.0 to LISP0.4100/192.2.101.70
139896: Jul 16 18:26:17.173: MRT(2): RPF change 192.2.101.70/LISP0.4100 for
(192.2.31.3/32, 239.100.100.100)
```

At this point, a SPT cutover is triggered as well so that the last hop router is connected to the first hop router via the shortest path tree. The last hop router triggers this (Edge2, in our case), as can be seen from the below logs from Edge2:

Aninda Chatterjee (aninchat@cisco.com)

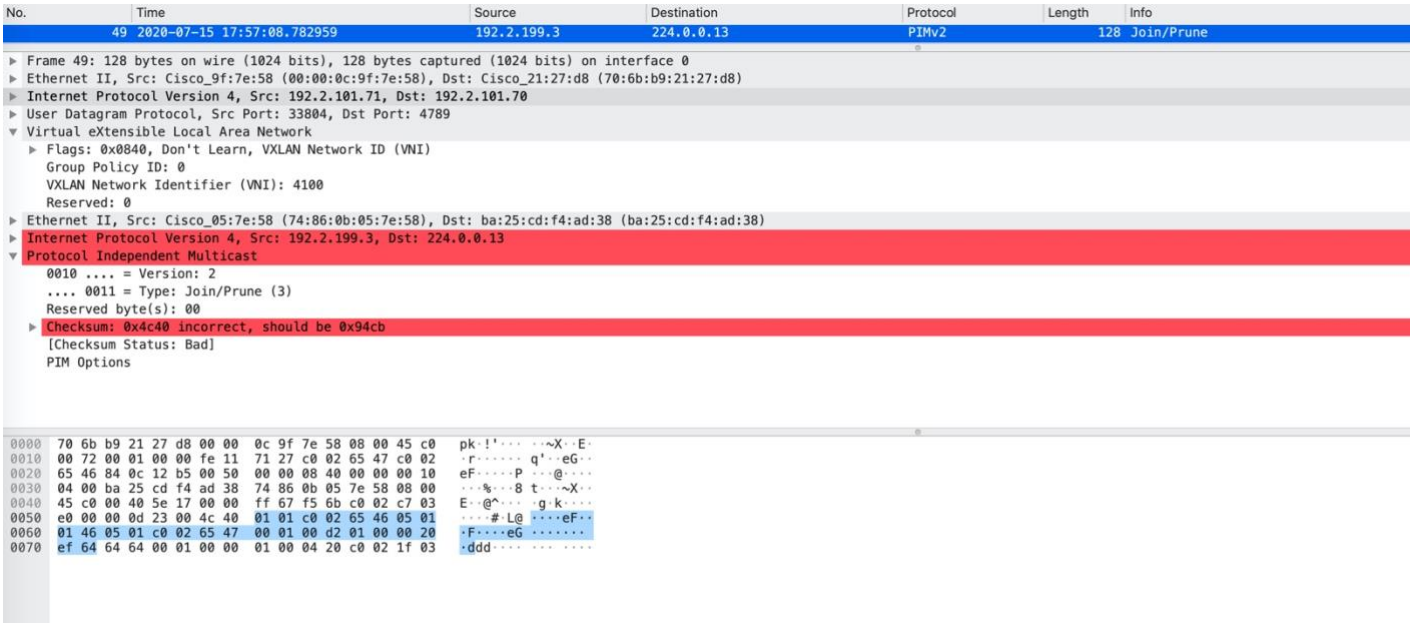
```
// trigger SPT cutover, send S-bit join to source RLOC

139893: Jul 16 18:26:17.172: MRT(2): Set the T-flag for (192.2.31.3, 239.100.100.100)
139897: Jul 16 18:26:17.173: MRT(2): Reset the F-flag for (192.2.31.3, 239.100.100.100)
139905: Jul 16 18:26:17.173: PIM(2): Insert (192.2.31.3,239.100.100.100) join in nbr
192.2.101.70's queue
139943: Jul 16 18:26:17.176: PIM(2): Building Join/Prune packet for nbr 192.2.101.70
139944: Jul 16 18:26:17.176: PIM(2):  Adding v2 (192.2.31.3/32, 239.100.100.100), S-bit
Join
139945: Jul 16 18:26:17.176: PIM(2): Adding LISP Unicast transport attribute in
join/prune to 192.2.101.70 (LISP0.4100)
139946: Jul 16 18:26:17.176: PIM(2): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```

The (S,G) join is sent to 192.2.101.70, as seen from the above logs. This is the RLOC IP address that was resolved via route watch earlier.



Aninda Chatterjee (aninchat@cisco.com)

It is important to look at this packet because it contains a lot of good information (timestamp might be off because the debugs were taken after the captures):



As you can see, the packet is generated by 192.2.199.3 (which is Edge2s loopback configured for multicast for Corp_VN). This is sent to Edge1. Because of the PIM join attributes for LISP, Wireshark is unable to decode the packet fully (lack of dissectors), however, we can dump the raw hex data and get a lot of good information from it.

```
0000   70 6b b9 21 27 d8 00 00 0c 9f 7e 58 08 00 45 c0
0010   00 72 00 01 00 00 fe 11 71 27 c0 02 65 47 c0 02
0020   65 46 84 0c 12 b5 00 50 00 00 08 40 00 00 00 10
0030   04 00 ba 25 cd f4 ad 38 74 86 0b 05 7e 58 08 00
0040   45 c0 00 40 5e 17 00 00 ff 67 f5 6b c0 02 c7 03
0050   e0 00 00 0d 23 00 4c 40 01 01 c0 02 65 46 05 01
0060   01 46 05 01 c0 02 65 47 00 01 00 d2 01 00 00 20
0070   ef 64 64 64 00 01 00 00 01 00 04 20 c0 02 1f 03
```

23 – PIM version of 2 and type of 3 (implying a join/prune).
c0 02 65 46 – 192.2.101.70 (the destination this join is meant for).
05 01 01 – this is the transport attribute. The first 8 bits give you the fbit, the ebit and the type. The first bit is the fbit, the second bit is the ebit and the last 6 bits form the type of attribute (should be set to 5 to indicate it is the transport attribute). So in this case, we have 0000 0101 which gives us a fbit of 0, an ebit of 0 and a type of 5. The next 8 bits specify the length which is 01 in this case. **The last 8 bits specify the transport – this is very important because this dictates whether the underlay is using multicast or headend replication (unicast)**. This is set to 0 for multicast and 1 for unicast. In our case, as you can see, it is 01, which means head-end replication.
46 05 01 - this is the receiver RLOC attribute. This is primarily used to provide the RLOC IP address of the ETR. The first 8 bits are again used for fbit, ebit and the type. In this case, we have a hex of 46 which translates to 0100 0110. The first bit is the fbit which is 0, the second bit is the ebit which is 1 (implying an end of attributes and that no more attributes follow), the last 6 bits form the type which is 000110, giving us a value of 6, which is the receiver RLOC attribute. The next 8 bits are used for length (the value is in octets and is used to specify the length of the actual attribute value + 1 to account for address family). It is 5 because the RLOC value is an IP address (4 octets) and 1 octet will be used for the address family. Thus, 01 is the address family which is IPv4 (address family numbers can be found on

the official IANA page and PIM simply follows that till 127 - https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml). **c0 02 65 47** – this is the RLOC address. Decoding this, you get 192.2.101.71, which is the loopback0 IP address of Edge2.

This (S,G) join is processed on Edge1:

```
// Edge1 processing S-bit join from Edge2

018249: Jul 16 18:26:17.177: PIM(2): J/P Transport Attribute, Transport Type: Unicast
018250: Jul 16 18:26:17.177: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
018251: Jul 16 18:26:17.177: MRT(2): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 192.2.101.71 Outer-source: 0.0.0.0 Successful
018260: Jul 16 18:26:17.177: MRT(2): Add LISP0.4100/192.2.101.71 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
018261: Jul 16 18:26:17.177: PIM(2): Add LISP0.4100/192.2.101.71 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join
```


Aninda Chatterjee (aninchat@cisco.com)

Final MRIB state on all devices

Edge1:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 03:29:39/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 03:29:39/00:02:01, flags: FT
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.71, Forward/Sparse, 03:29:38/00:02:32
```

Border1:

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1d04h/00:02:38, RP 192.2.199.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.71, Forward/Sparse, 1d04h/00:02:38

(192.2.31.3, 239.100.100.100), 03:30:27/00:02:42, flags: P
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70
  Outgoing interface list: Null
```

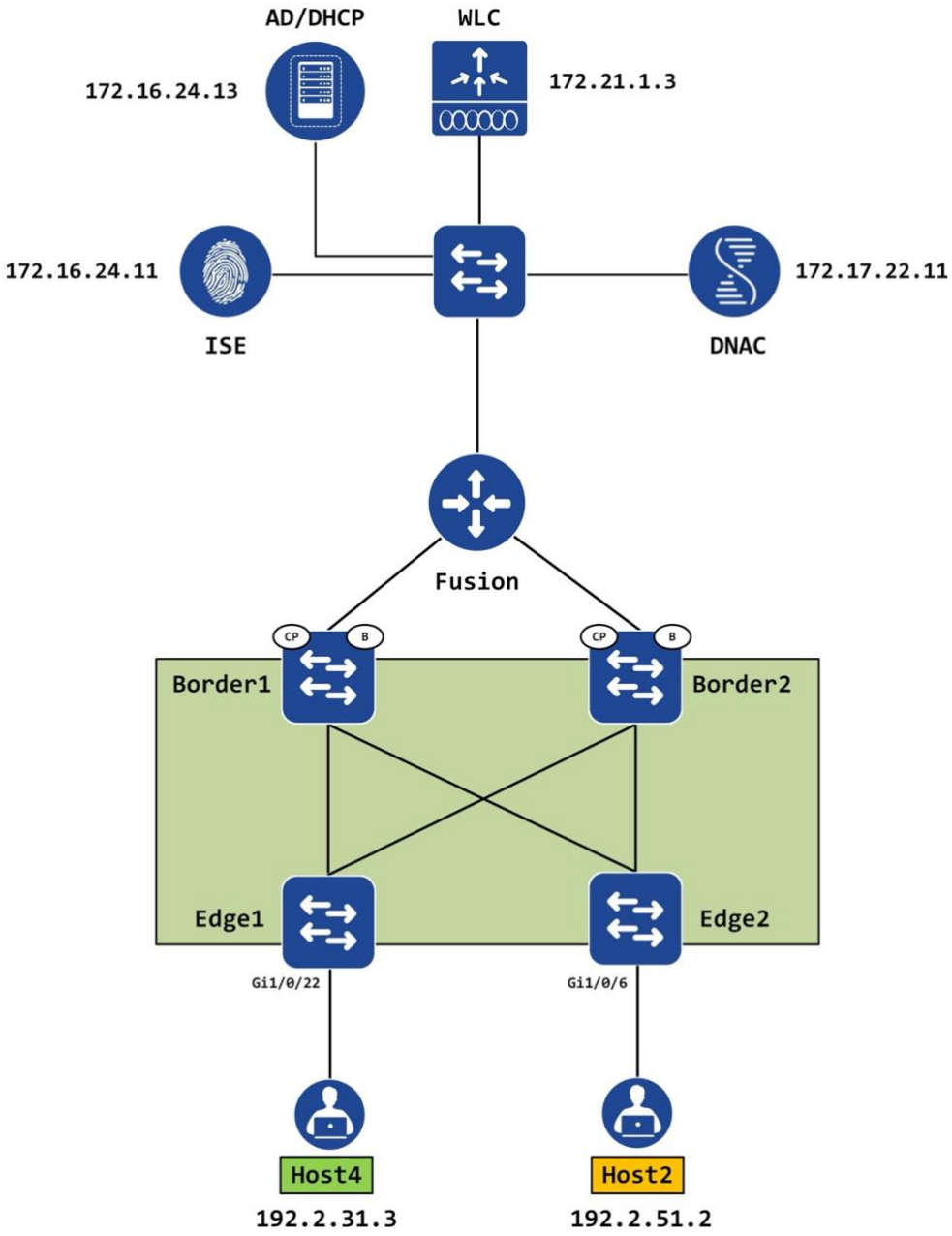
Edge2:

```
Edge2#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1d04h/stopped, RP 192.2.199.1, flags: SJC
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 1d04h/00:02:22

(192.2.31.3, 239.100.100.100), 03:30:55/00:02:18, flags: JT
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 03:30:55/00:02:22
```


Aninda Chatterjee (aninchat@cisco.com)

Head-end replication - Source in fabric, receiver outside fabric

The topology for this is the same as before, but the receiver is now beyond the fusion device (essentially a loopback on the switch to which ISE/DNAC etc are connected).



Since the receiver is outside the fabric and there maybe a number of devices before Border1 (RP), none of this will be automated by DNAC. It is important to ensure that all devices in the path are configured correctly with RPs IP address.

The SVI that extends the VRF towards the Fusion must have PIM enabled under it and you should have an active PIM neighbor peering over the SVI.

Aninda Chatterjee (aninchat@cisco.com)

```
Border1#show run int vlan 3014
Building configuration...

Current configuration : 204 bytes
!
interface Vlan3014
 description vrf interface to External router
 vrf forwarding Corp_VN
 ip address 192.2.100.1 255.255.255.252
 no ip redirects
 ip pim sparse-mode
 ip route-cache same-interface
end
```

```
Border1#show ip pim vrf Corp_VN 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
192.2.100.2       Vlan3014                 1d22h/00:01:20    v2    1 / DR S P G
```

All of the devices must be correctly configured with the RP IP address as well:

```
Shared_Services_SW#show ip pim rp mapping 239.100.100.100
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 192.2.199.1 (?)

Fusion#show ip pim rp mapping 239.100.100.100
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 192.2.199.1 (?)
```


Aninda Chatterjee (aninchat@cisco.com)

Building the shared tree

Loopback0 on the shared services switch is configured to join the multicast group 239.100.100.100.

```
Shared_Services_SW#show run int lo0
Building configuration...

Current configuration : 119 bytes
!
interface Loopback0
 ip address 99.99.99.99 255.255.255.255
 ip pim passive
 ip igmp join-group 239.100.100.100
end
```

We should now see this in the IGMP groups and membership table as well.

```
Shared_Services_SW#show ip igmp groups 239.100.100.100
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter   Group
Accounted
239.100.100.100  Loopback0                00:01:04  00:02:17  99.99.99.99

Shared_Services_SW#show ip igmp membership 239.100.100.100
Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP, the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude

 Channel/Group                 Reporter        Uptime   Exp.  Flags  Interface
 *,239.100.100.100             99.99.99.99     00:01:21 02:59 2LA    Lo0
```

This should also trigger PIM (*,G) joins to be sent for this group, hop by hop, all the way till the RP (Border1).

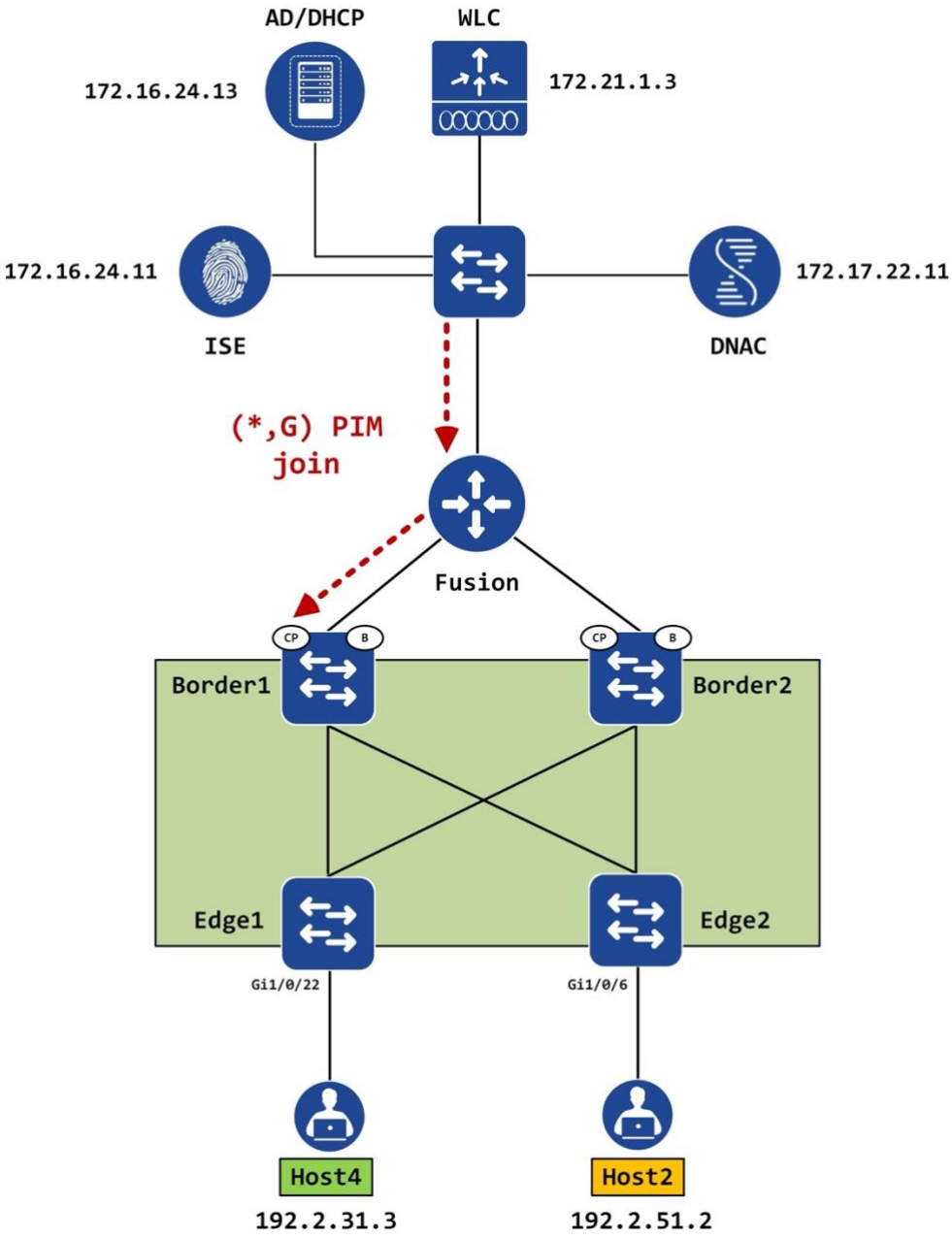
Border1 should receive this and build the appropriate state in its multicast routing table. Remember, Border1 will receive this on an interface in the VRF Corp_VN and thus, the state will be built in the multicast routing table for this VRF.

```
// PIM (*,G) join received on Border1

Jul 17 15:21:39.951: PIM(117): Received v2 Join/Prune on Vlan3014 from 192.2.100.2, to
us
Jul 17 15:21:39.951: PIM(117): Join-list: (*, 239.100.100.100), RPT-bit set, WC-bit
set, S-bit set
```


Aninda Chatterjee (aninchat@cisco.com)

```
// Border1 creates entry in MRIB

Jul 17 15:21:39.952: MRIB: Table = VRF Corp_VN, Trans: (*, 239.100.100.100) mroute
update: ADD, rpf = Null
Jul 17 15:21:39.954: PIM(117): Check RP 192.2.199.1 into the (*, 239.100.100.100) entry
Jul 17 15:21:39.956: MRT(117): RPF Track start on 192.2.199.1 for (*, 239.100.100.100)
Jul 17 15:21:39.958: PIM(117): Adding register decap tunnel (Tunnel3) as accepting
interface of (*, 239.100.100.100).
Jul 17 15:21:39.959: MRT(117): Create (*,239.100.100.100), RPF (unknown, 0.0.0.0, 0/0)
Jul 17 15:21:39.960: MRT(117): WAVL Insert interface: Vlan3014 in (* ,239.100.100.100)
Successful
Jul 17 15:21:39.963: MRT(117): Add Vlan3014/239.100.100.100 to the olist of (*,
239.100.100.100), Forward state - MAC not built
```

The process would look like this:



Border1 should now have a (*,G) entry for this group:

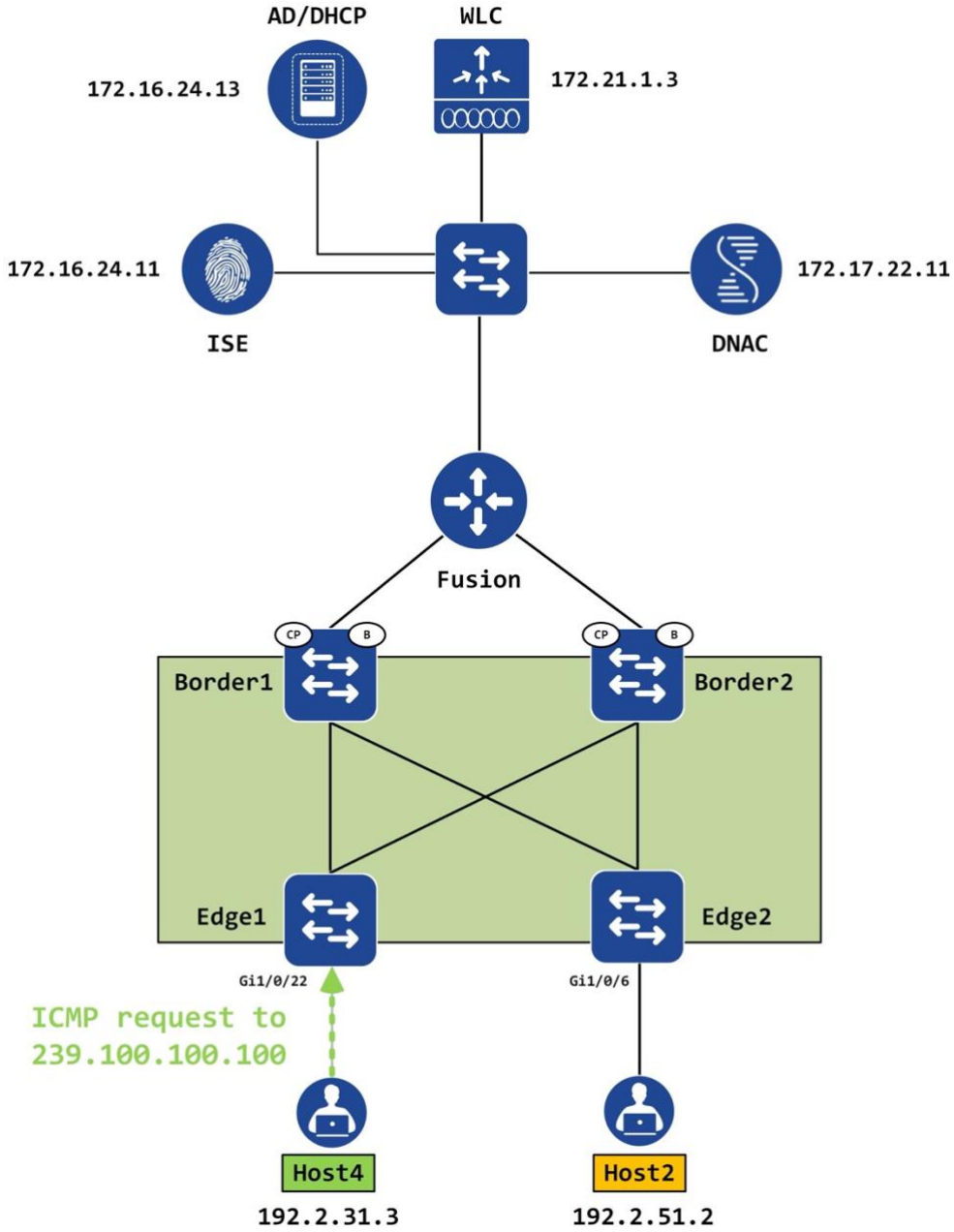
```
Border1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 04:22:23/00:03:02, RP 192.2.199.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan3014, Forward/Sparse, 04:22:23/00:03:02
```

The incoming interface is Null and RPF neighbor is 0.0.0.0 because Border1 is the RP itself. The outgoing interface list includes VLAN3014 which is the L3 interface on which the (*,G) join was received.

Aninda Chatterjee (aninchat@cisco.com)

Start of source stream

Host4 starts to send traffic to 239.100.100.100 – in this case, it is a simple ping (ICMP request).



On Edge1, the packet first gets punted to create state in the multicast routing table.

```
018885: Jul 18 11:46:29.059: FMANRP-MCAST: (192.2.31.3, 239.100.100.100) mfib entry not
found
018886: Jul 18 11:46:29.059: FMANRP-MCAST: (*, 239.100.100.100) mfib entry not found
018887: Jul 18 11:46:29.059: FMANRP-MCAST: Punted packet from Vlan1029 tagged.
018888: Jul 18 11:46:29.079: FMANRP-MCAST(2): add (192.2.31.3, 239.100.100.100/64), id
0x0
```


Aninda Chatterjee (aninchat@cisco.com)

```
// MRIB state created

018900: Jul 18 11:46:29.080: MRIB: Table = VRF Corp_VN, [4  ] i/o - received 36 bytes
018901: Jul 18 11:46:29.080: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Received   DDE* [Vl1029  SP* ]
018902: Jul 18 11:46:29.080: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Updated   DDE* #A=0, #F=0 [Vl1029  SP* ]
018903: Jul 18 11:46:29.080: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Added
018932: Jul 18 11:46:29.083: MRT(2): Create (*,239.100.100.100), RPF (unknown, 0.0.0.0,
0/0)

// rwatch triggered to resolve source and RPF track started for source

018934: Jul 18 11:46:29.083: MCAST-RWATCH: -af IPv4 -topo 8002: Route watch topology
found.
018935: Jul 18 11:46:29.083: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix not found.
018936: Jul 18 11:46:29.083: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix created.
018937: Jul 18 11:46:29.083: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in RIB.
018938: Jul 18 11:46:29.083: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in LISP.
018939: Jul 18 11:46:29.083: MRT(2): RPF Track start on 192.2.31.3 for (192.2.31.3,
239.100.100.100)

// RPF updated for (*,G)

018947: Jul 18 11:46:29.083: MRT(2): (192.2.31.3,239.100.100.100), RPF install from
/0.0.0.0 to Vlan1029/0.0.0.0
018953: Jul 18 11:46:29.084: MRT(2): RPF change 0.0.0.0/Vlan1029 for (192.2.31.3/32,
239.100.100.100)
```

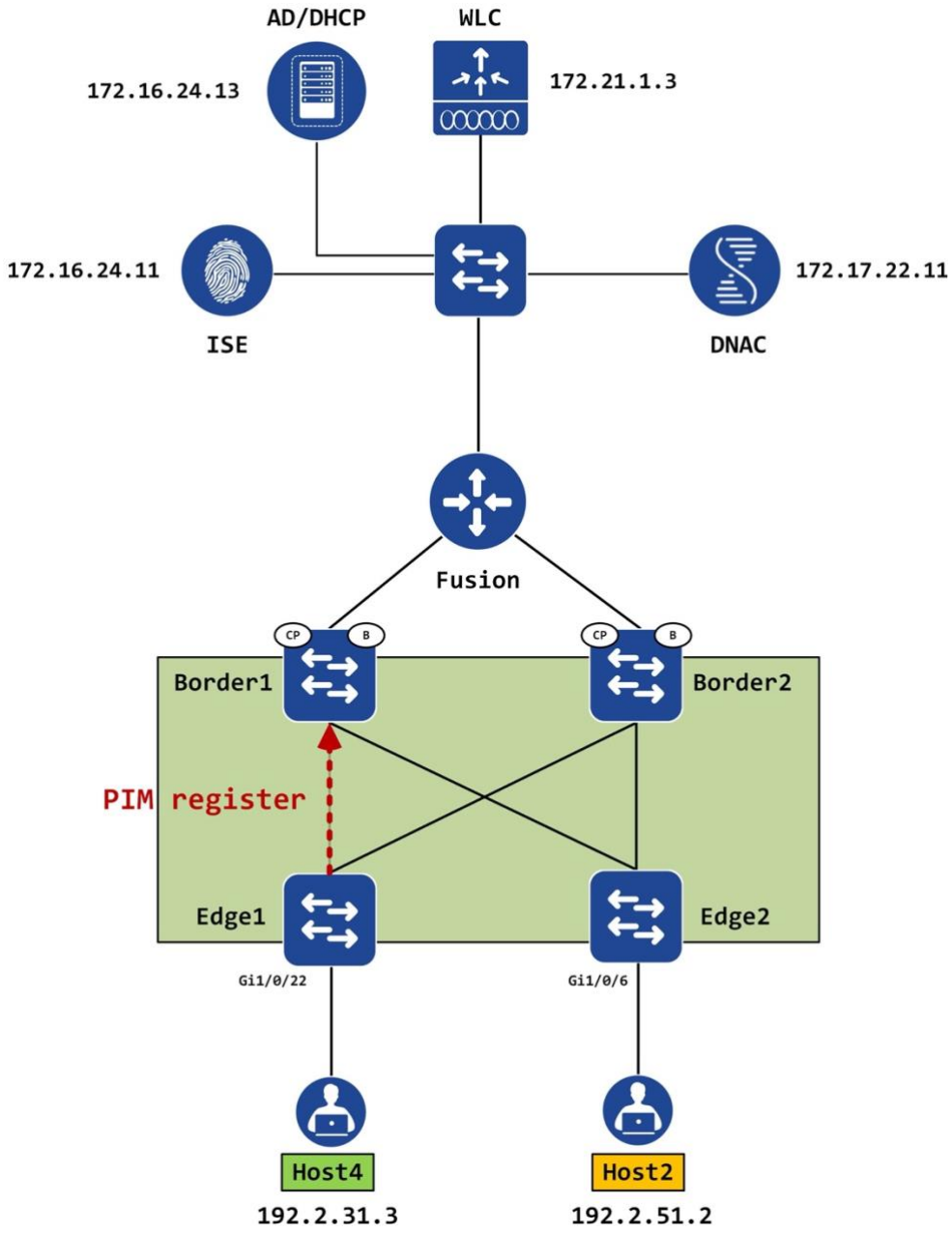
Edge1 should now register the source with the RP (remember, this is still ASM in the overlay). It creates a register encap tunnel, encapsulates the multicast data packets into a unicast packet for the RP:

```
018957: Jul 18 11:46:29.084: PIM(2): Adding register encap tunnel (Tunnel2) as
forwarding interface of (192.2.31.3, 239.100.100.100).
018958: Jul 18 11:46:29.084: MRT(2): Set the F-flag for (*, 239.100.100.100)
018959: Jul 18 11:46:29.084: MRT(2): Set the F-flag for (192.2.31.3, 239.100.100.100)
018967: Jul 18 11:46:29.085: MRT(2): Set the T-flag for (192.2.31.3, 239.100.100.100)
```

Border1 (RP) gets this register:

```
Jul 18 11:46:33.589: PIM(117): Received v2 Register on LISP0.4100 from 192.2.199.2
Jul 18 11:46:33.590:      for 192.2.31.3, group 239.100.100.100
```


Aninda Chatterjee (aninchat@cisco.com)



We can also see the following packet capture for this:

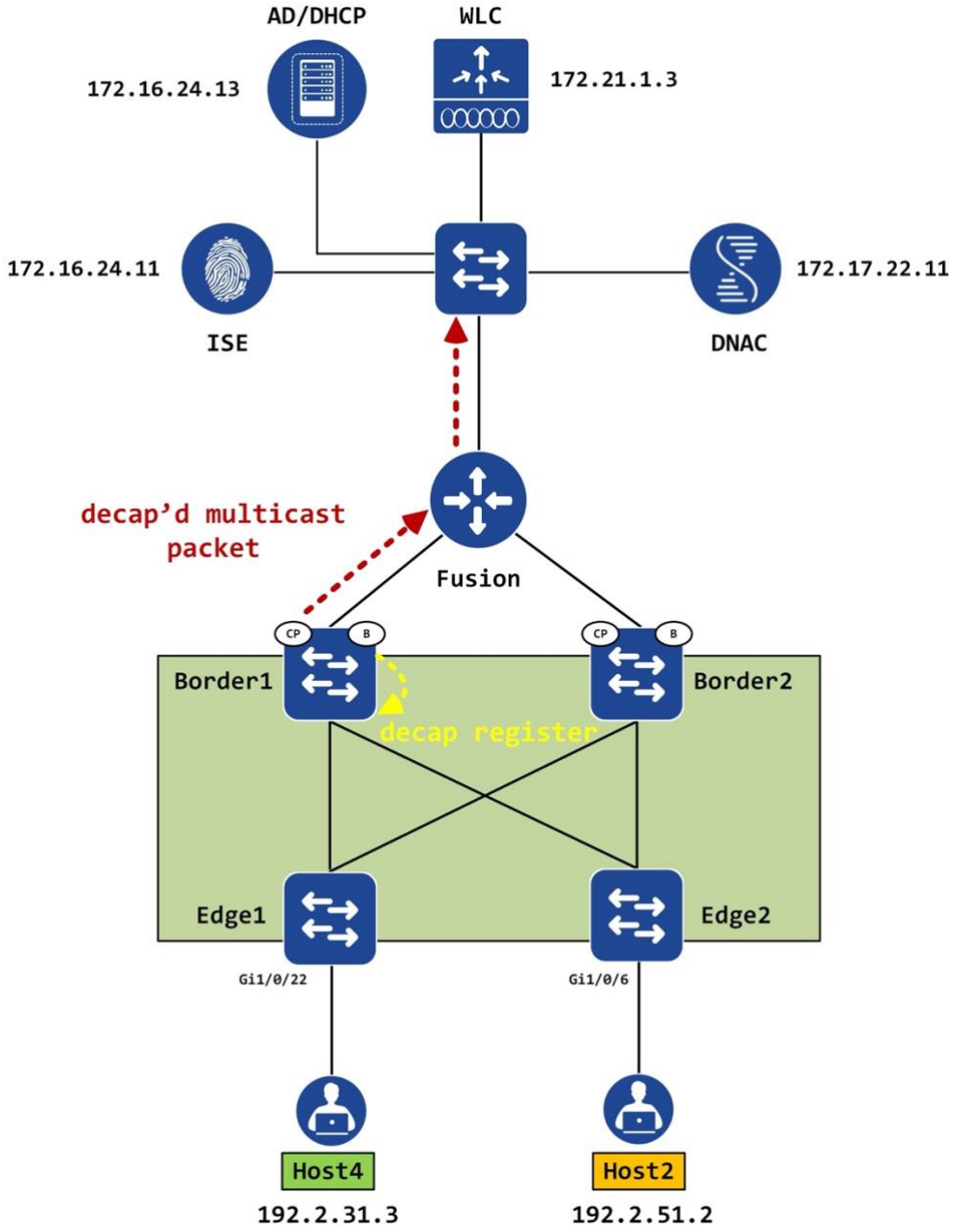
No.	Time	Source	Destination	Protocol	Length	Info
36	2020-07-18 11:46:33.589272	192.2.31.3	239.100.100.100	PIMv2	152	Register

```
▶ Frame 36: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface 0
▶ Ethernet II, Src: Cisco_9f:1f:d6 (00:00:0c:9f:1f:d6), Dst: Cisco_21:27:e4 (70:6b:b9:21:27:e4)
▶ Internet Protocol Version 4, Src: 192.2.101.70, Dst: 192.2.101.65
▶ User Datagram Protocol, Src Port: 26371, Dst Port: 4789
▼ Virtual eXtensible Local Area Network
  ▶ Flags: 0x0840, Don't Learn, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 4100
    Reserved: 0
▶ Ethernet II, Src: Cisco_05:1f:d6 (74:86:0b:05:1f:d6), Dst: ba:25:cd:f4:ad:38 (ba:25:cd:f4:ad:38)
▶ Internet Protocol Version 4, Src: 192.2.199.2, Dst: 192.2.199.1
▼ Protocol Independent Multicast
    0010 .... = Version: 2
    .... 0001 = Type: Register (1)
    Reserved byte(s): 00
    Checksum: 0xdeff [correct]
    [Checksum Status: Good]
  ▶ PIM Options
▶ Internet Protocol Version 4, Src: 192.2.31.3, Dst: 239.100.100.100
▶ Internet Control Message Protocol
```

The actual ICMP packet is encapsulated inside a PIM register. This is in turn encapsulated in a VXLAN header and sent to Border1.

Aninda Chatterjee (aninchat@cisco.com)

Like before, several things now happen on Border1: the packet is decapsulated, and sent towards the receiver using the shared tree.



Next, (S,G) state is also built on the border. The outgoing interface list is copied from the (*,G) into the (S,G):

```
Jul 18 11:46:33.594: PIM(117): Adding register decap tunnel (Tunnel3) as accepting
interface of (192.2.31.3, 239.100.100.100).
Jul 18 11:46:33.596: MRT(117): Create (192.2.31.3,239.100.100.100), RPF (unknown,
0.0.0.0, 0/0)
Jul 18 11:46:33.600: MRT(117): Add Vlan3014/239.100.100.100 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
Jul 18 11:46:33.632: MRT(117): RPF change 192.2.101.70/LISP0.4100 for (192.2.31.3/32,
239.100.100.100)
```

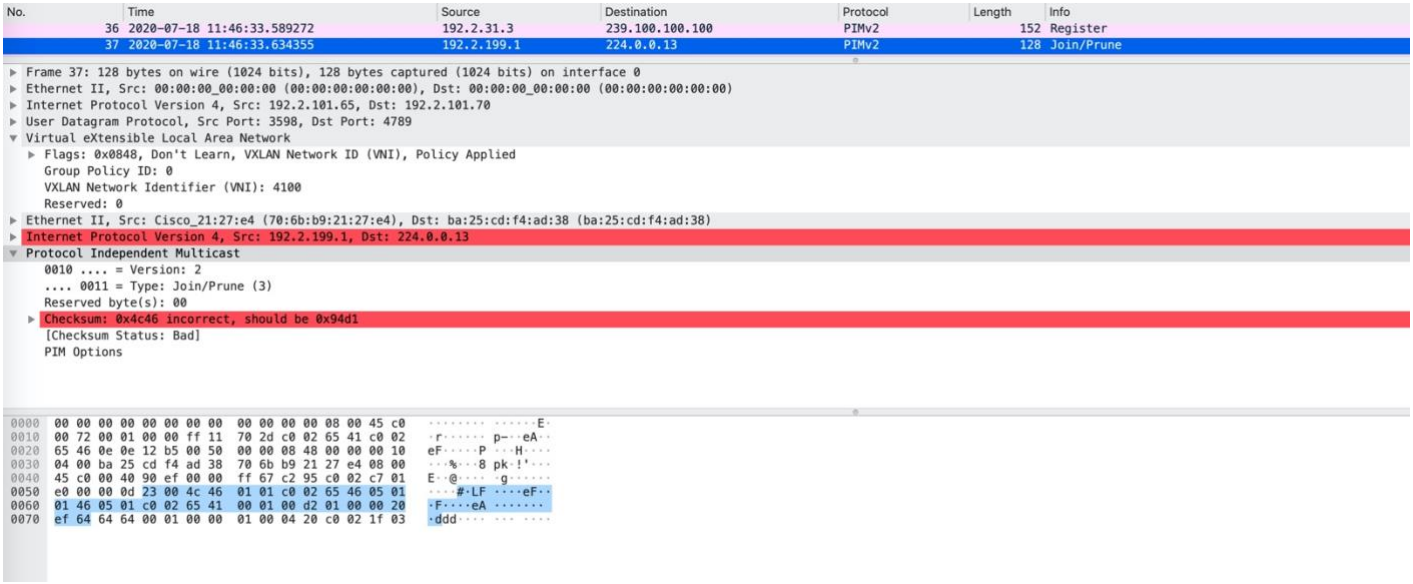

Aninda Chatterjee (aninchat@cisco.com)

Building the source tree

Once the (S,G) state is built on Border1 (the RP), it also triggers a (S,G) join to be sent towards the source to build the source tree, hop by hop.

```
Jul 18 11:46:33.632: PIM(117): Insert (192.2.31.3,239.100.100.100) join in nbr
192.2.101.70's queue
Jul 18 11:46:33.633: PIM(117): Building Join/Prune packet for nbr 192.2.101.70
Jul 18 11:46:33.633: PIM(117):  Adding v2 (192.2.31.3/32, 239.100.100.100), S-bit Join
Jul 18 11:46:33.633: PIM(117): Adding LISP Unicast transport attribute in join/prune to
192.2.101.70 (LISP0.4100)
Jul 18 11:46:33.633: PIM(117): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```

This packet is very important because it also informs Edge1 of the type of transport that will be used (multicast vs unicast). Because Wireshark cannot decode this, the raw hex dump needs to be manually analyzed, just like we did in the earlier section.



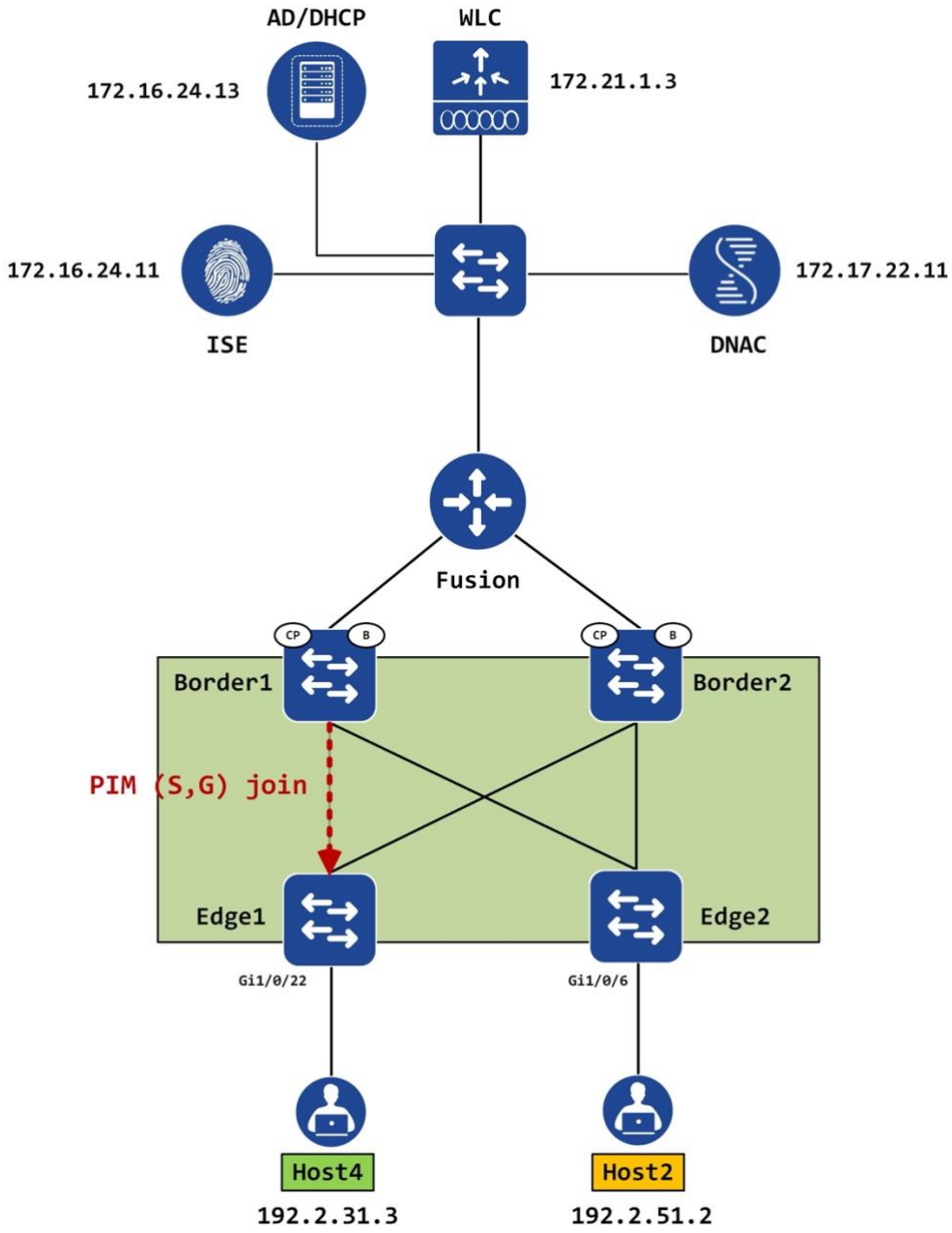
The hex dump is:

```
0000   00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 c0
0010   00 72 00 01 00 00 ff 11 70 2d c0 02 65 41 c0 02
0020   65 46 0e 0e 12 b5 00 50 00 00 08 48 00 00 00 10
0030   04 00 ba 25 cd f4 ad 38 70 6b b9 21 27 e4 08 00
0040   45 c0 00 40 90 ef 00 00 ff 67 c2 95 c0 02 c7 01
0050   e0 00 00 0d 23 00 4c 46 01 01 c0 02 65 46 05 01
0060   01 46 05 01 c0 02 65 41 00 01 00 d2 01 00 00 20
0070   ef 64 64 64 00 01 00 00 01 00 04 20 c0 02 1f 03
```

We're not going to go into the entire thing again however, the highlighted part (in yellow) confirms that the transport is unicast (implying head-end replication).

Since the Edge1 is directly connected to Border1, the (S,G) is directly sent to it – however, if there were intermediate nodes, every node will build this (S,G) state hop by hop and forward it towards the source.

Aninda Chatterjee (aninchat@cisco.com)



Edge1 gets this PIM join and updates the state in its multicast routing table:

```
019119: Jul 18 11:46:33.634: PIM(2): J/P Transport Attribute, Transport Type: Unicast
019120: Jul 18 11:46:33.634: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
019121: Jul 18 11:46:33.635: MRT(2): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 192.2.101.65 Outer-source: 0.0.0.0 Successful
019130: Jul 18 11:46:33.635: MRT(2): Add LISP0.4100/192.2.101.65 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
019131: Jul 18 11:46:33.635: PIM(2): Add LISP0.4100/192.2.101.65 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join
```


Aninda Chatterjee (aninchat@cisco.com)

SPT cutover

We're not going to go into too much detail here – just like the first section (source and receiver in fabric), when the decapsulated register reaches the last-hop router (the shared services switch, in this case), it will trigger a SPT cutover.

Since the shortest path to the source is via Border1, it will be a part of the SPT and the multicast routing tables will not change much here.

Final MRIB state on each device

Edge1:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 06:02:00/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 06:02:00/00:02:50, flags: FT
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.65, Forward/Sparse, 06:02:00/00:03:26
```

Border1:

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1d04h/00:02:30, RP 192.2.199.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan3014, Forward/Sparse, 1d04h/00:02:30

(192.2.31.3, 239.100.100.100), 06:02:35/00:02:29, flags: T
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70
  Outgoing interface list:
    Vlan3014, Forward/Sparse, 06:02:35/00:03:19
```


Aninda Chatterjee (aninchat@cisco.com)

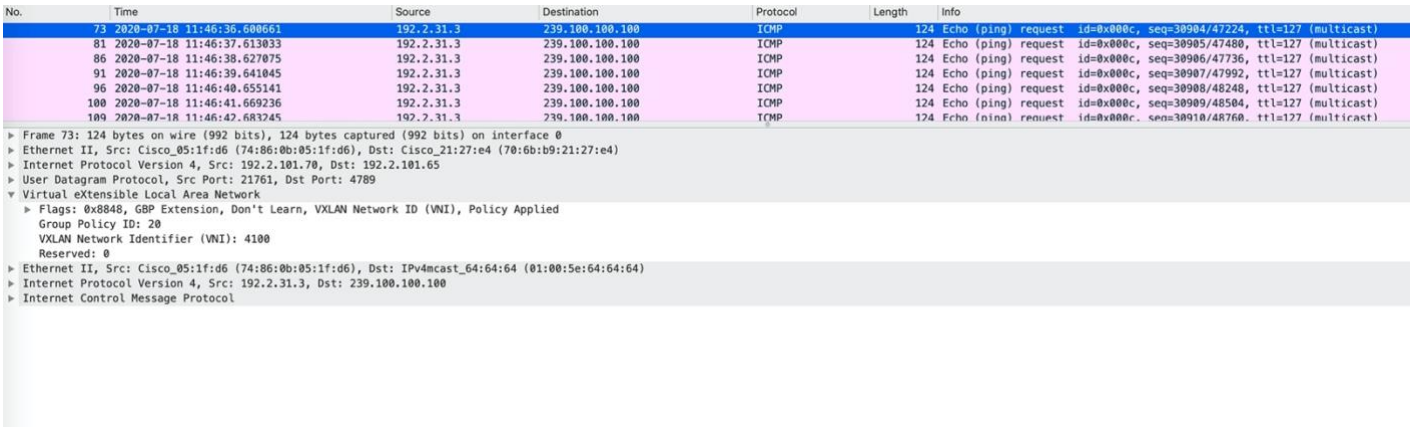
Fusion:

```
Fusion#show ip mroute 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1d04h/00:02:58, RP 192.2.199.1, flags: S
  Incoming interface: Gi0/0/1.3014, RPF nbr 192.2.100.1
  Outgoing interface list:
    Te0/0/1.100, Forward/Sparse, 1d04h/00:02:58

(192.2.31.3, 239.100.100.100), 06:03:18/00:03:24, flags: T
  Incoming interface: Gi0/0/1.3014, RPF nbr 192.2.100.1
  Outgoing interface list:
    Te0/0/1.100, Forward/Sparse, 06:03:18/00:03:17
```

Shared services switch:

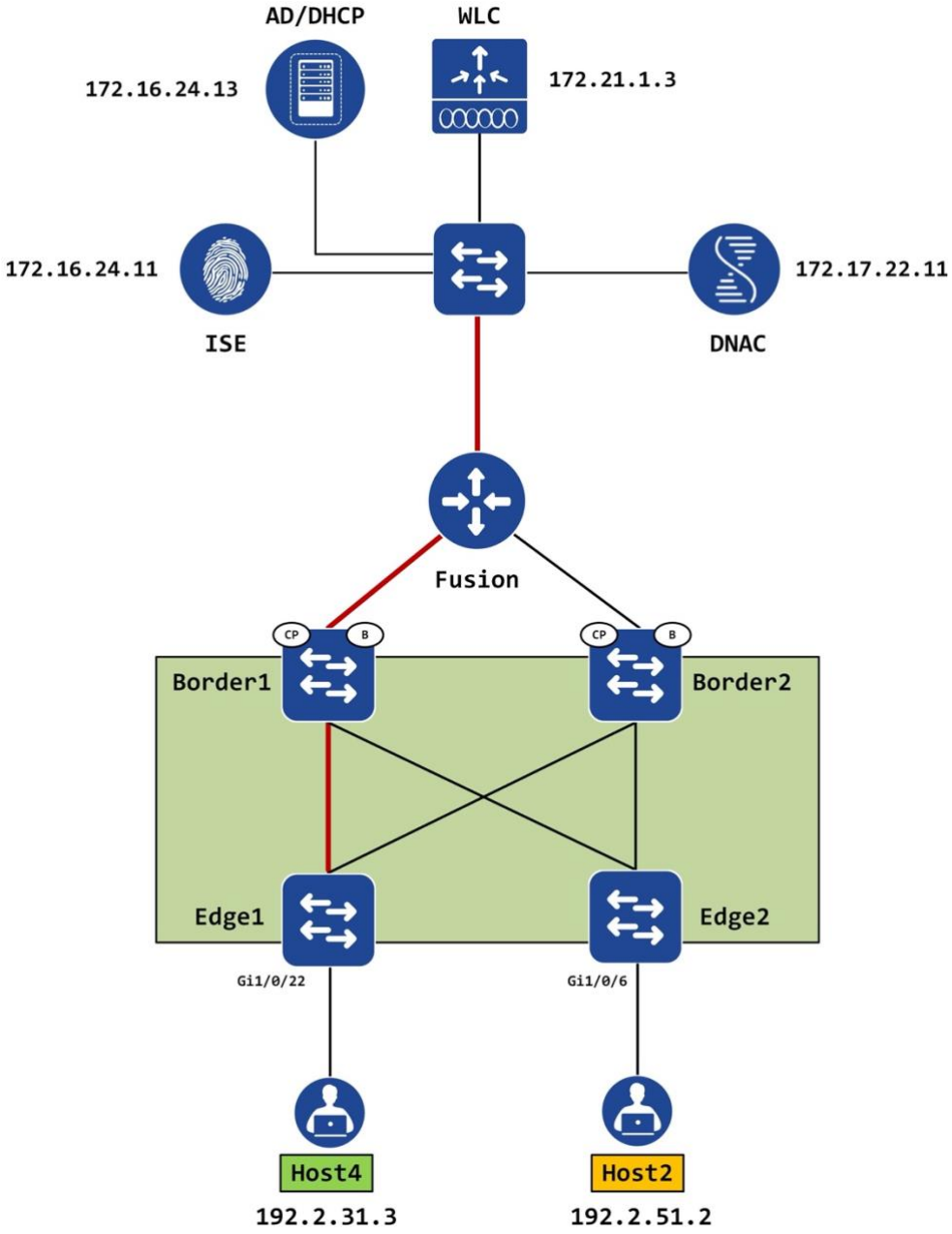
```
Shared_Services_SW#show ip mroute 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1d04h/stopped, RP 192.2.199.1, flags: SJCL
  Incoming interface: Vlan100, RPF nbr 172.18.1.1
  Outgoing interface list:
    Loopback0, Forward/Sparse-Dense, 1d04h/00:02:46

(192.2.31.3, 239.100.100.100), 06:05:59/00:02:08, flags: LJT
  Incoming interface: Vlan100, RPF nbr 172.18.1.1
  Outgoing interface list:
    Loopback0, Forward/Sparse-Dense, 06:05:59/00:02:46
```

A packet capture of the multicast packet in the fabric after SPT cutover:



The final multicast tree:

Aninda Chatterjee (aninchat@cisco.com)



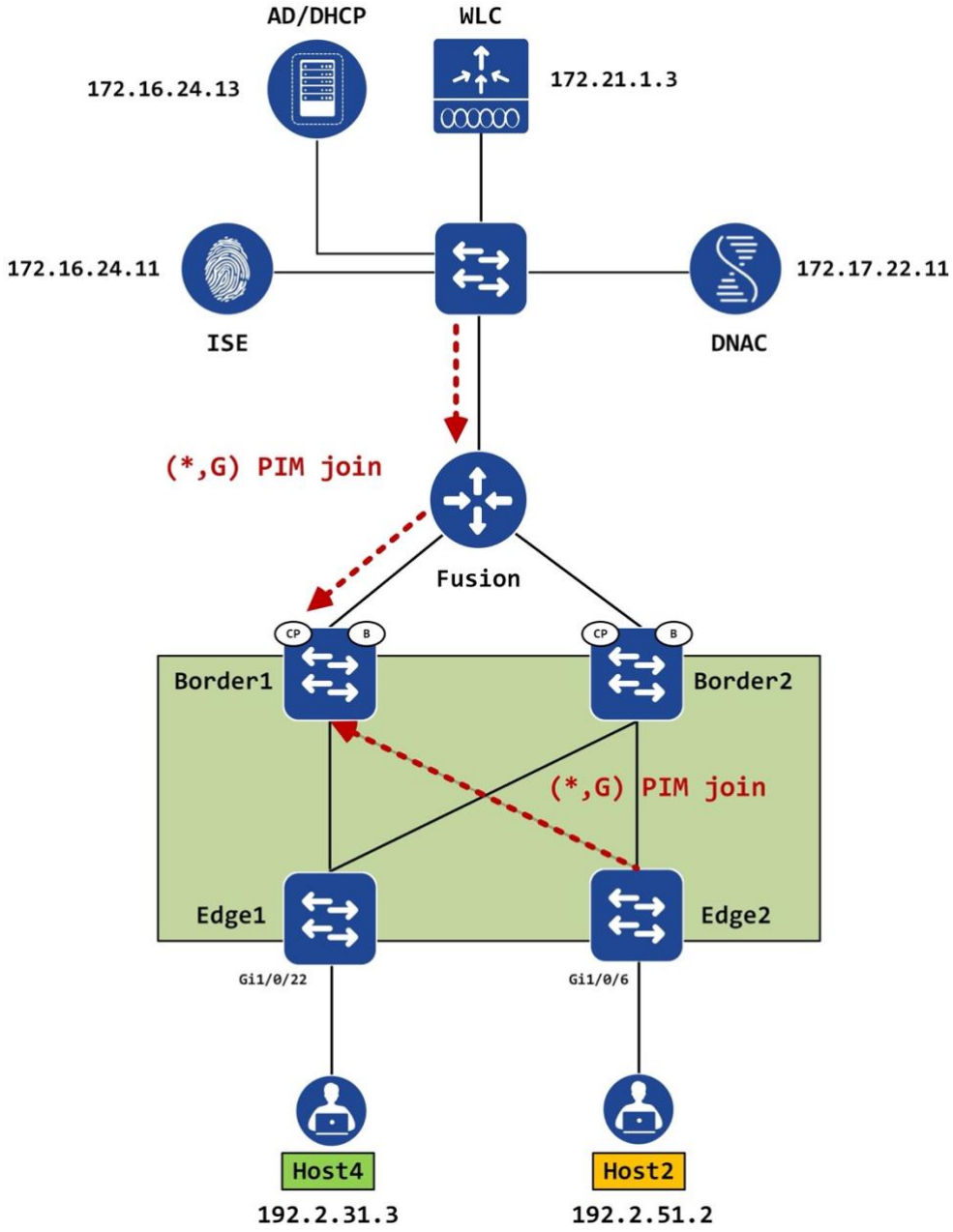
Aninda Chatterjee (aninchat@cisco.com)

Head-end replication - Source in fabric, receiver in and outside fabric

We're not going to go into too much detail since most of the concepts are the same.

Building the shared tree

In this case, we have a receiver in the fabric and outside the fabric. Both of these receivers will send IGMP joins (membership reports) to 239.100.100.100 and these last-hop routers will send (*,G) PIM joins for this group to the RP (Border1). Border1, thus, should be receiving this from both outside the fabric and from inside the fabric:



Border1 should now have its multicast routing table with both interfaces in the outgoing interface list:

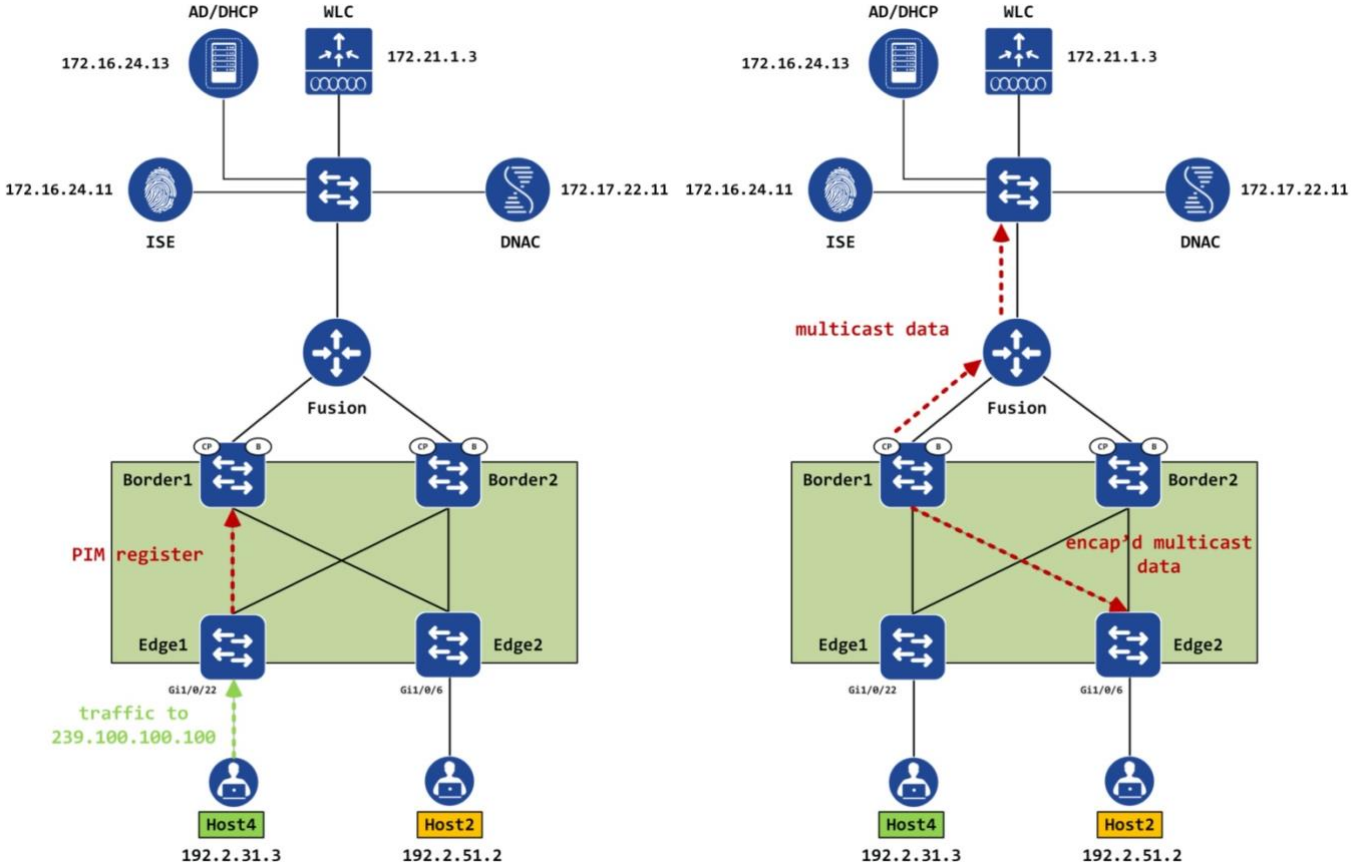
```
Border1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 2d21h/00:02:41, RP 192.2.199.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.71, Forward/Sparse, 02:58:48/00:02:41
    Vlan3014, Forward/Sparse, 2d21h/00:02:32
```


Aninda Chatterjee (aninchat@cisco.com)

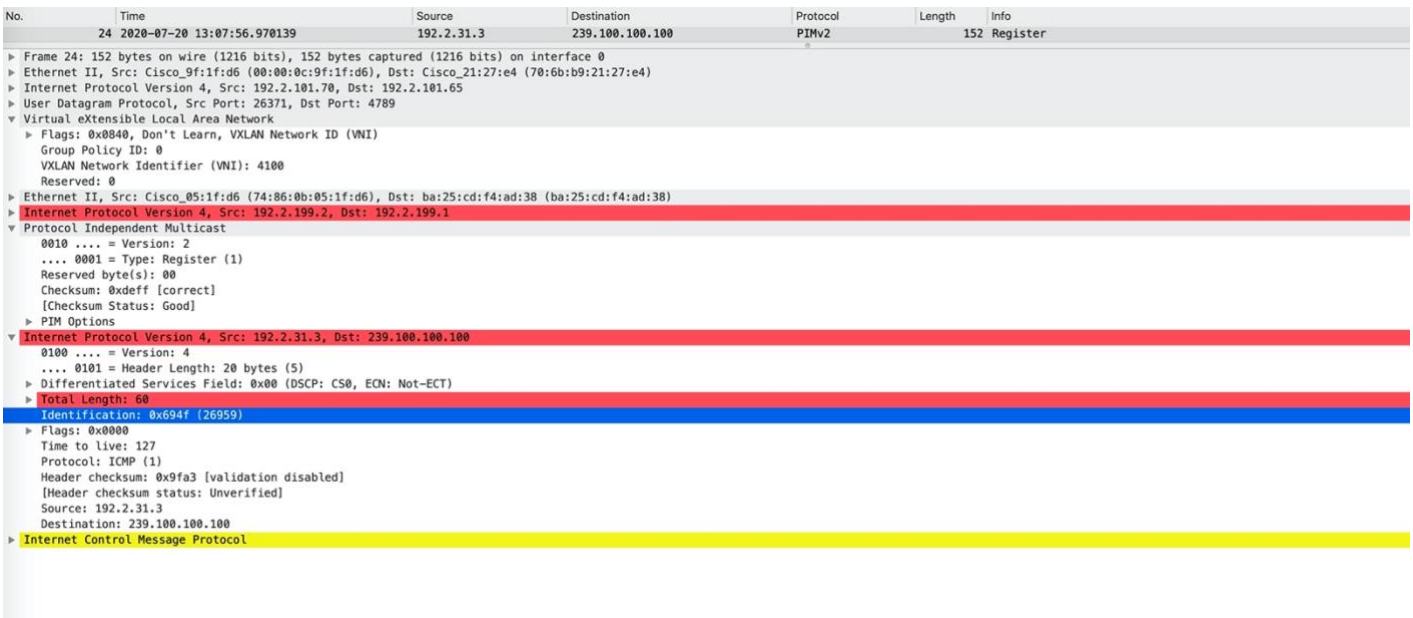
Start of source stream

Host4 starts to send traffic to 239.100.100.100 now. Again, like before, the source is first registered to the RP (Border1). Once the register is decapsulated, the packet is replicated and sent in both directions – to the receiver outside the fabric and to the receiver inside the fabric.

This entire flow can be visualized as the follows:



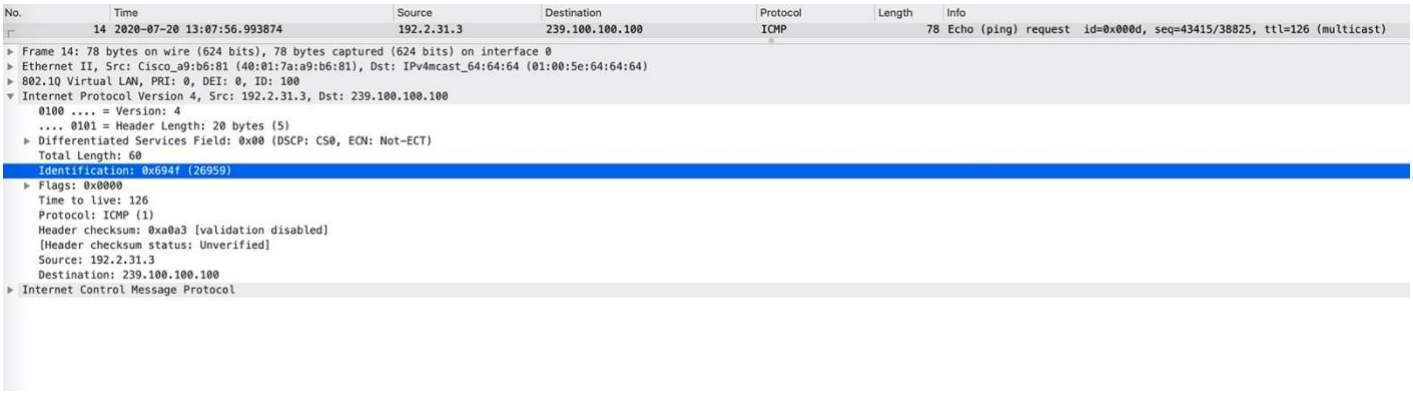
Consider the register coming to Border1:



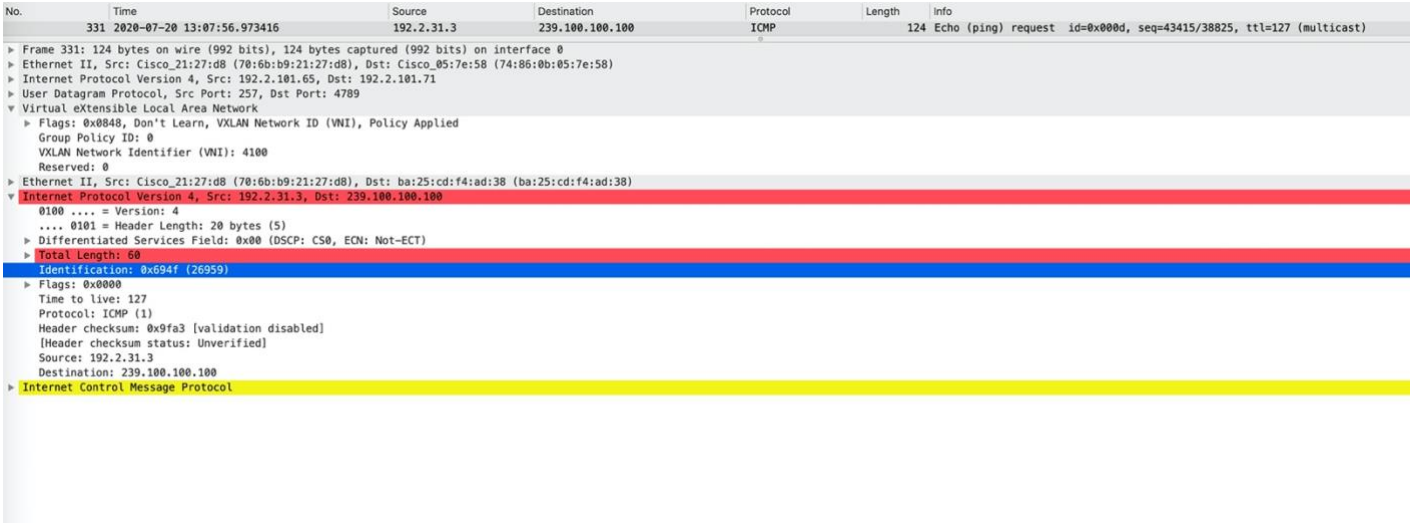
Aninda Chatterjee (aninchat@cisco.com)

Carefully notice the IP ID – **0x694F**. This is the same ID that should be used (post decapsulation on Border1), when the packet is replicated and sent to the receiver outside the fabric and the receiver inside the fabric.

Let’s confirm that. The following is a packet capture on the shared services switch:



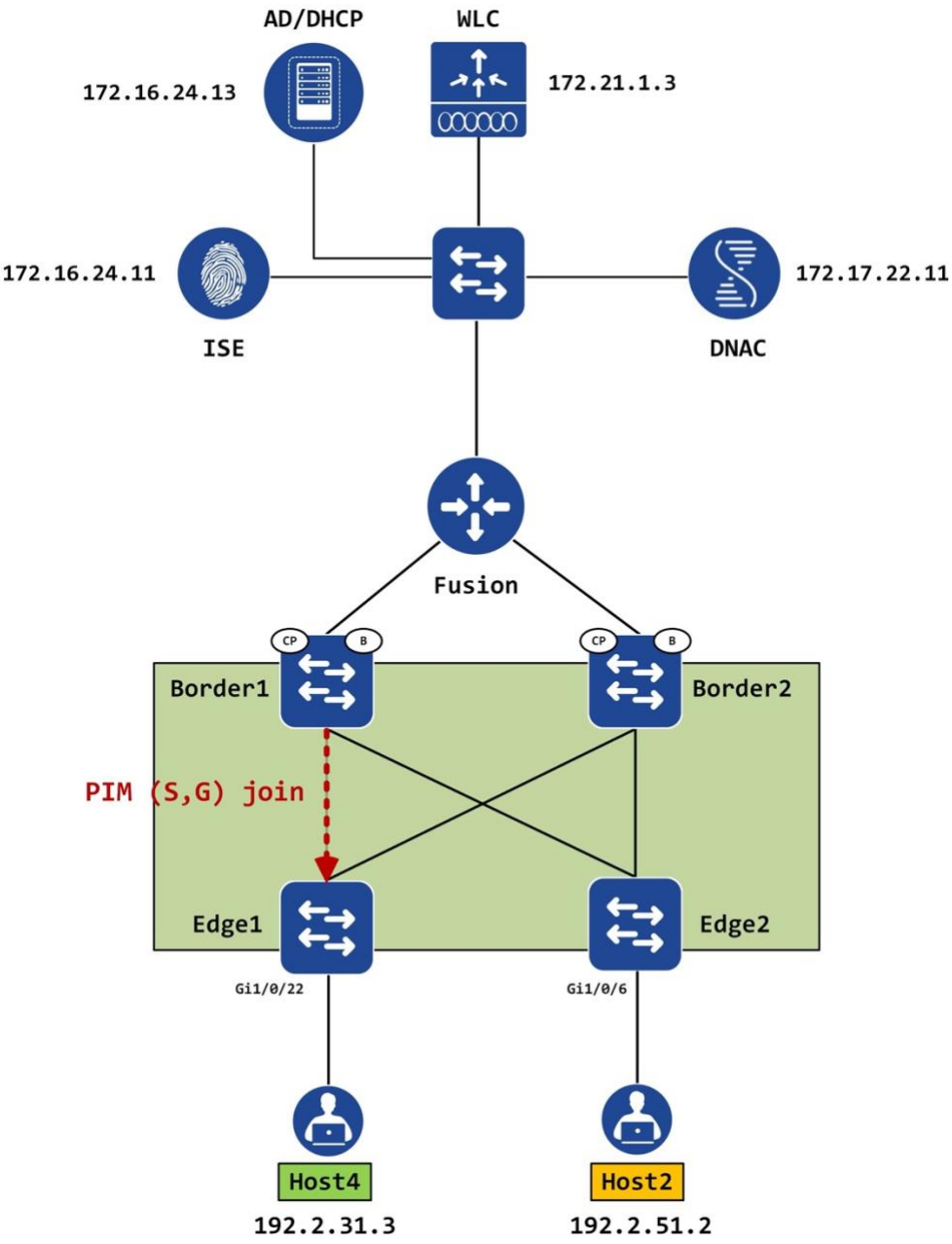
And the following is a packet capture on Edge2:



As can be seen from all three packets, the IP ID remains 0x694F. The reason that the packet captured on Edge2 is encapsulated again is because it is within the fabric and needs to be VXLAN encapsulated.

Building the source tree

The building of the source tree is the same as the previous sections. The register receipt on Border1 leads to the creation of a (S,G) entry in the multicast routing table, which in turn triggers a (S,G) PIM join towards the source.

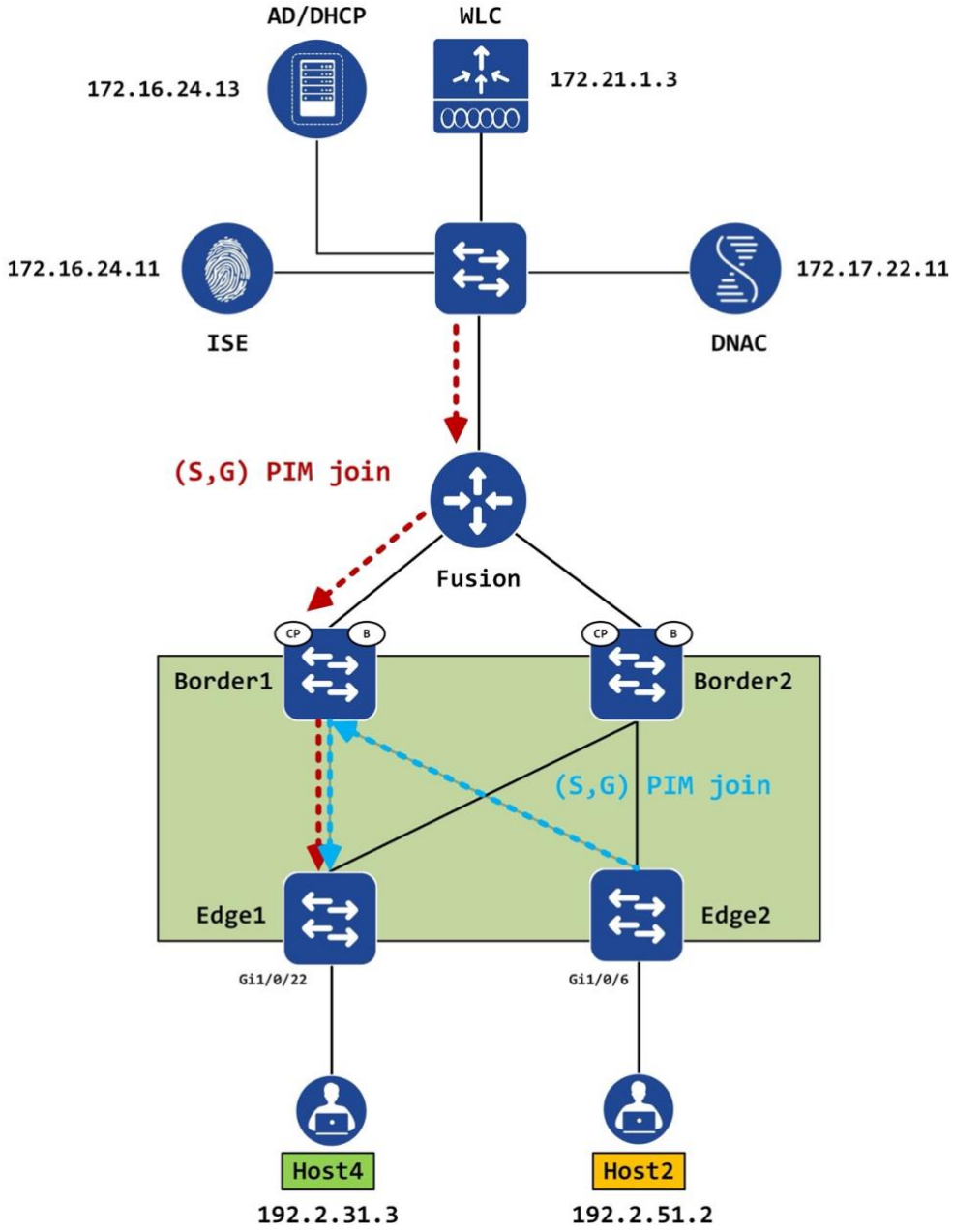


Aninda Chatterjee (aninchat@cisco.com)

SPT cutover

The SPT cutover is triggered by the last-hop routers and the process is the same as already documented. The only difference is that we have two last-hop routers in this case – the shared services switch (outside the fabric) and Edge2 (inside the fabric).

Both of these will send (S,G) PIM joins with S-bit set towards the source with every hop in the middle building this (S,G) entry. For the shared services switch, when the (S,G) reaches Border1, it will send this join towards the source with the LISP join attributes inside the PIM message (this is important because it determines what transport we are using inside the fabric – multicast or unicast).



The following debugs show Edge1 receiving both these joins:

```
// Edge1 receiving join from Border1

005656: Jul 20 13:45:03.092: PIM(2): J/P Transport Attribute, Transport Type: Unicast
005657: Jul 20 13:45:03.092: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
005658: Jul 20 13:45:03.092: MRT(2): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 192.2.101.65 Outer-source: 0.0.0.0 Successful
005666: Jul 20 13:45:03.092: MRT(2): Add LISP0.4100/192.2.101.65 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
005667: Jul 20 13:45:03.092: PIM(2): Add LISP0.4100/192.2.101.65 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join

//Edge1 receiving join from Edge2

005618: Jul 20 13:45:03.058: PIM(2): J/P Transport Attribute, Transport Type: Unicast
005619: Jul 20 13:45:03.058: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
005620: Jul 20 13:45:03.058: MRT(2): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 192.2.101.71 Outer-source: 0.0.0.0 Successful
005629: Jul 20 13:45:03.058: MRT(2): Add LISP0.4100/192.2.101.71 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
005630: Jul 20 13:45:03.058: PIM(2): Add LISP0.4100/192.2.101.71 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join
```

The multicast routing table on Edge1 correctly shows both interfaces in the outgoing interface list:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:00:09/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 00:00:09/00:03:22, flags: FT
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.65, Forward/Sparse, 00:00:08/00:03:21
    LISP0.4100, 192.2.101.71, Forward/Sparse, 00:00:08/00:03:21
```


Aninda Chatterjee (aninchat@cisco.com)

Final MRIB state on each device

Edge1:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 00:29:21/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 00:29:21/00:02:10, flags: FT
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.65, Forward/Sparse, 00:29:21/00:02:41
    LISP0.4100, 192.2.101.71, Forward/Sparse, 00:29:21/00:02:41
```

Border1:

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 2d23h/00:03:29, RP 192.2.199.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, 192.2.101.71, Forward/Sparse, 04:54:51/00:03:29
    Vlan3014, Forward/Sparse, 2d23h/00:02:39

(192.2.31.3, 239.100.100.100), 00:29:28/00:02:37, flags: T
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70
  Outgoing interface list:
    Vlan3014, Forward/Sparse, 00:29:28/00:02:39
```

Edge2:

```
Edge2#show ip mroute vrf Corp_VN 239.100.100.100
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 05:02:51/stopped, RP 192.2.199.1, flags: SJC
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 05:02:51/00:02:56

(192.2.31.3, 239.100.100.100), 00:37:28/00:02:14, flags: JT
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 00:37:28/00:02:56
```


Aninda Chatterjee (aninchat@cisco.com)

An important thing to note here – Border1 does not show both interfaces in the outgoing interface list - this is because (as you will see in the case of native multicast as well), Border1 gets pruned from the multicast tree within the fabric since it only needs to forward based on the outer header.

```
Jul 20 13:45:03.128: PIM(117): J/P Transport Attribute, Transport Type: Unicast
Jul 20 13:45:03.128: PIM(117): Join-list: (*, 239.100.100.100), RPT-bit set, WC-bit
set, S-bit set
Jul 20 13:45:03.130: PIM(117): Prune-list: (192.2.31.3/32, 239.100.100.100) RPT-bit set
Jul 20 13:45:03.130: MRT(117): Start 0xFFA0AB37E0 [2390] expire timer interface:
LISP0.4100 (192.2.31.3, 239.100.100.100) delay 180 sec
Jul 20 13:45:03.130: PIM(117): Prune LISP0.4100/192.2.101.71 from (192.2.31.3/32,
239.100.100.100)
Jul 20 13:45:03.131: MRT(117): Delete LISP0.4100/192.2.101.71 from the olist of
(192.2.31.3, 239.100.100.100)
```


Aninda Chatterjee (aninchat@cisco.com)

Implications on IP adjacency table usage

IP adjacencies play an important role when dealing with head-end replication. Even though the packet is multicast, the transport is unicast and thus IP adjacencies will be built appropriately for this as well.

Prior to starting the source, the following adjacency table usage exists:

```
Edge1#show platform hardware fed switch active fwd-asic resource features ip-adjacency
utilization
IPv4 unicast adjacency resource info
Resource Info for ASIC Instance: 0 [A:0, C:0]
Shared Resource Name                    Allocated       Free              Usage%
-------------------------------------------------------------------------------
RSC_RI                                  139             57181               0.24
RSC_SI                                  782             64586               1.20

Rewrite Data                            Allocated       Free              Usage%
-------------------------------------------------------------------------------
PHF_EGRESS_destMacAddress               5               31995               0.02

CAM Table Utilization Info              Allocated       Free              Usage%
-------------------------------------------------------------------------------
IP Route table Host/Network               196/ 317      24380/7875   0.80/  3.87

IPv4 unicast adjacency resource info
Resource Info for ASIC Instance: 1 [A:0, C:1]
Shared Resource Name                    Allocated       Free              Usage%
-------------------------------------------------------------------------------
RSC_RI                                  139             57181               0.24
RSC_SI                                  782             64586               1.20

Rewrite Data                            Allocated       Free              Usage%
-------------------------------------------------------------------------------
PHF_EGRESS_destMacAddress               5               31995               0.02

CAM Table Utilization Info              Allocated       Free              Usage%
-------------------------------------------------------------------------------
IP Route table Host/Network                  0/   0          0/0      0.00/  0.00
```

During this time, we only have an adjacency to 192.2.101.65:

```
Edge1#show platform software fed switch active ip adj | in 4100
225.0.0.0       Loopback4100            f800.0351.0000   0x7f0f3038ed08 0x0
0x0     0xf8000351  2020/07/18 13:34:24.206
192.2.101.65    LISP0.4100              4500.0000.0000   0x7f0f31d0af08 0x7f0f31d0b118
0x60    0x324       2020/07/18 13:34:48.455
227.0.0.0       LISP0.4100              f800.0394.0000   0x7f0f325e95a8 0x0
0x20    0xf8000394  2020/07/18 13:34:18.116
225.0.0.0       LISP0.4100              f800.0391.0000   0x7f0f325e9a98 0x0
0x20    0xf8000391  2020/07/18 13:34:18.116
```


Aninda Chatterjee (aninchat@cisco.com)

Once the source starts, take a look at the usage again:

```
Edge1#show platform hardware fed switch active fwd-asic resource features ip-adjacency
utilization
IPv4 unicast adjacency resource info
Resource Info for ASIC Instance: 0 [A:0, C:0]
Shared Resource Name                    Allocated       Free              Usage%
-------------------------------------------------------------------------------
RSC_RI                                  141             57179               0.25
RSC_SI                                  784             64584               1.20

Rewrite Data                            Allocated       Free              Usage%
-------------------------------------------------------------------------------
PHF_EGRESS_destMacAddress               5               31995               0.02

CAM Table Utilization Info              Allocated       Free              Usage%
-------------------------------------------------------------------------------
IP Route table Host/Network               196/ 317      24380/7875   0.80/  3.87

IPv4 unicast adjacency resource info
Resource Info for ASIC Instance: 1 [A:0, C:1]
Shared Resource Name                    Allocated       Free              Usage%
-------------------------------------------------------------------------------
RSC_RI                                  141             57179               0.25
RSC_SI                                  784             64584               1.20

Rewrite Data                            Allocated       Free              Usage%
-------------------------------------------------------------------------------
PHF_EGRESS_destMacAddress               5               31995               0.02

CAM Table Utilization Info              Allocated       Free              Usage%
-------------------------------------------------------------------------------
IP Route table Host/Network                  0/   0          0/0      0.00/  0.00
```

The usage increased by 2 entries. Also, as can be seen, we form a new adjacency for Edge2:

```
Edge1#show platform software fed switch active ip adj | in 4100
225.0.0.0       Loopback4100            f800.0351.0000   0x7f0f3038ed08 0x0
0x0      0xf8000351  2020/07/18 13:34:24.206
192.2.101.65    LISP0.4100              4500.0000.0000   0x7f0f31d0af08 0x7f0f31d0b118
0x60     0x324       2020/07/18 13:34:48.455
227.0.0.0       LISP0.4100              f800.0394.0000   0x7f0f325e95a8 0x0
0x20     0xf8000394  2020/07/18 13:34:18.116
192.2.101.71    LISP0.4100              4500.0000.0000   0x7f0f328b2168 0x7f0f328b2388
0x60     0x80        2020/07/20 13:07:56.990
225.0.0.0       LISP0.4100              f800.0391.0000   0x7f0f325e9a98 0x0
0x20     0xf8000391  2020/07/18 13:34:18.116
```

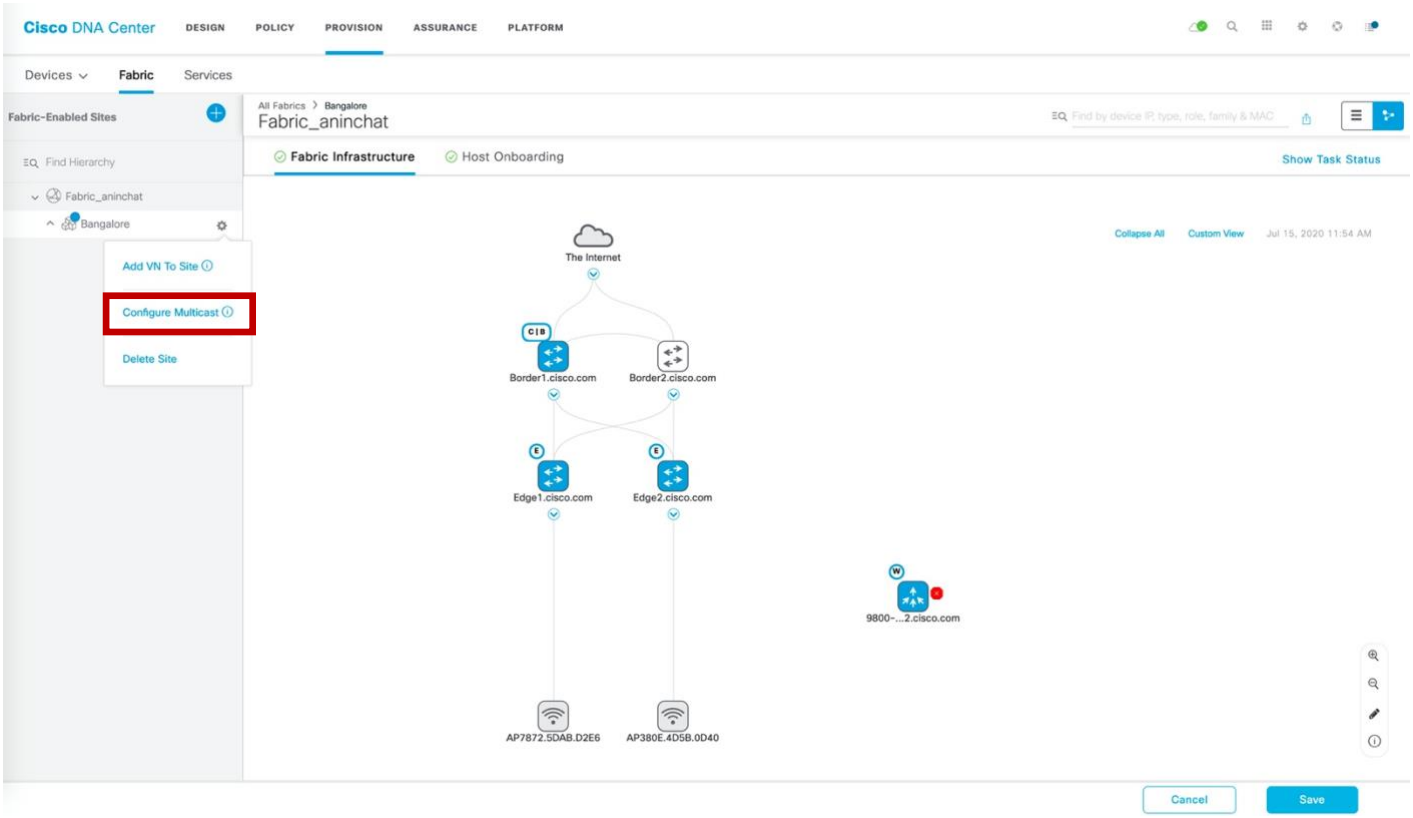
Thus, even if there is no unicast communication to a host behind Edge2, we still need to create an IP adjacency for Edge2 (on Edge1) for head-end replication. Such adjacencies are midchain adjacencies are increment the RSC_RI count by 2.

Aninda Chatterjee (aninchat@cisco.com)

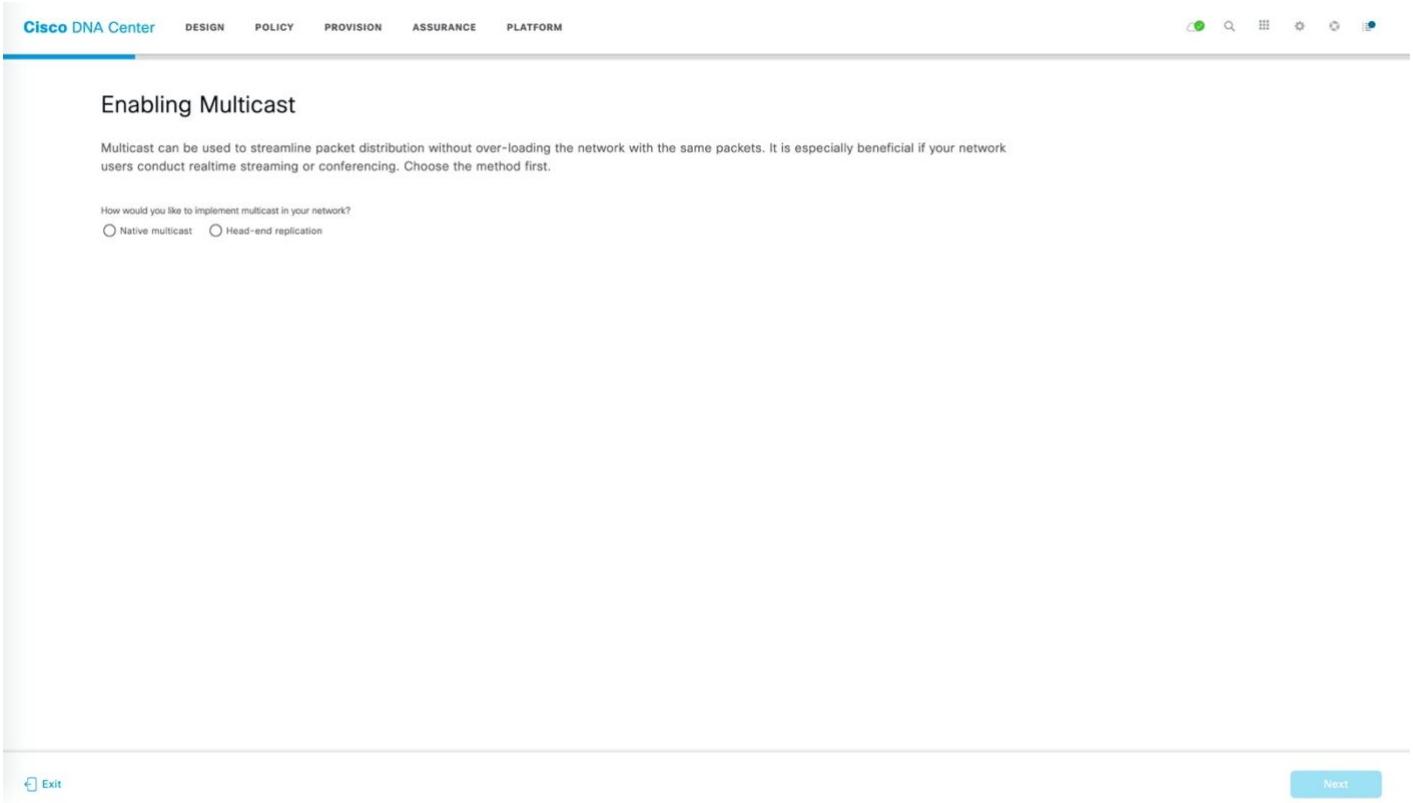
Native multicast

Configuring ASM in the overlay via DNAC

Go back and remove all multicast configuration. Traverse to your fabric page again and select ‘Configure Multicast’ once more:

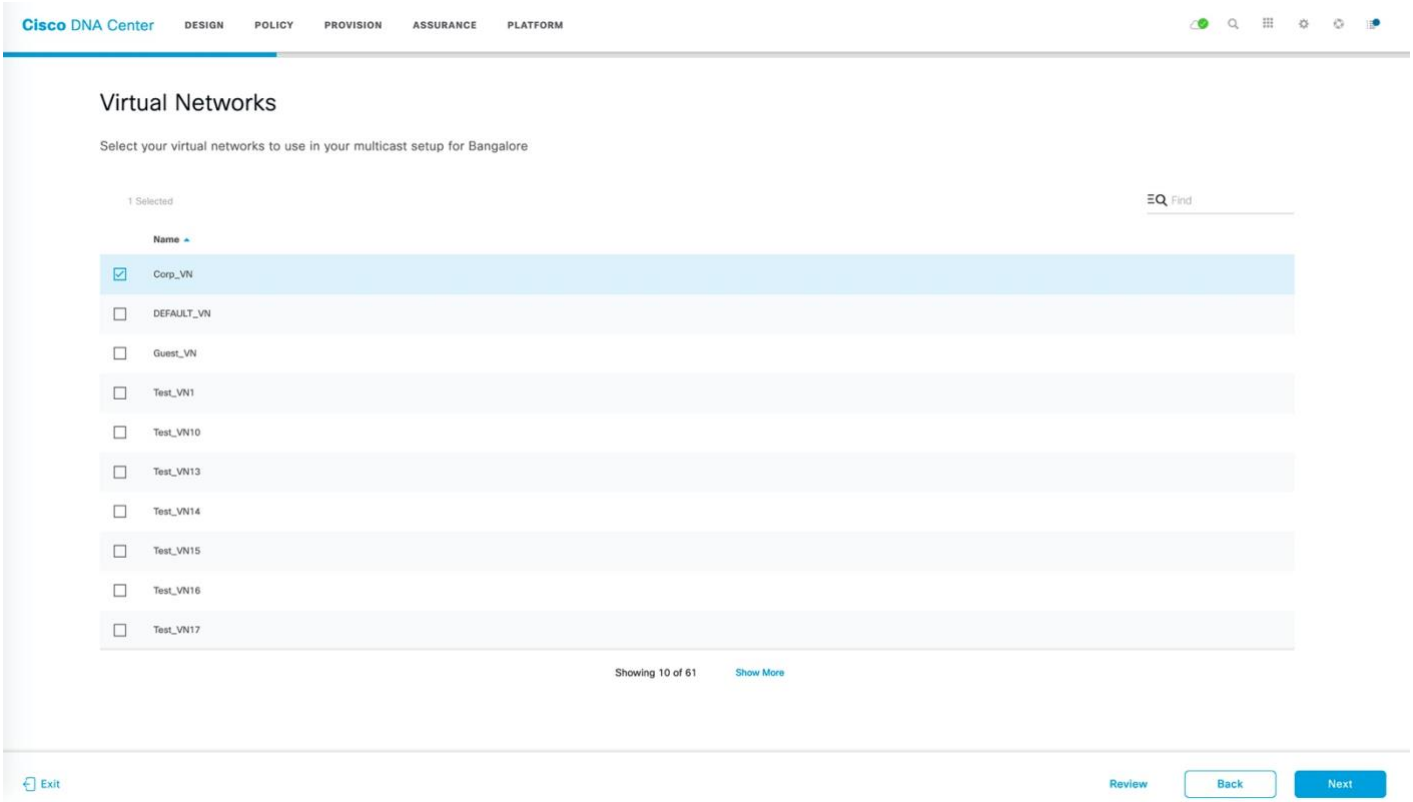


From here, choose native multicast or headend replication. We will choose native multicast.

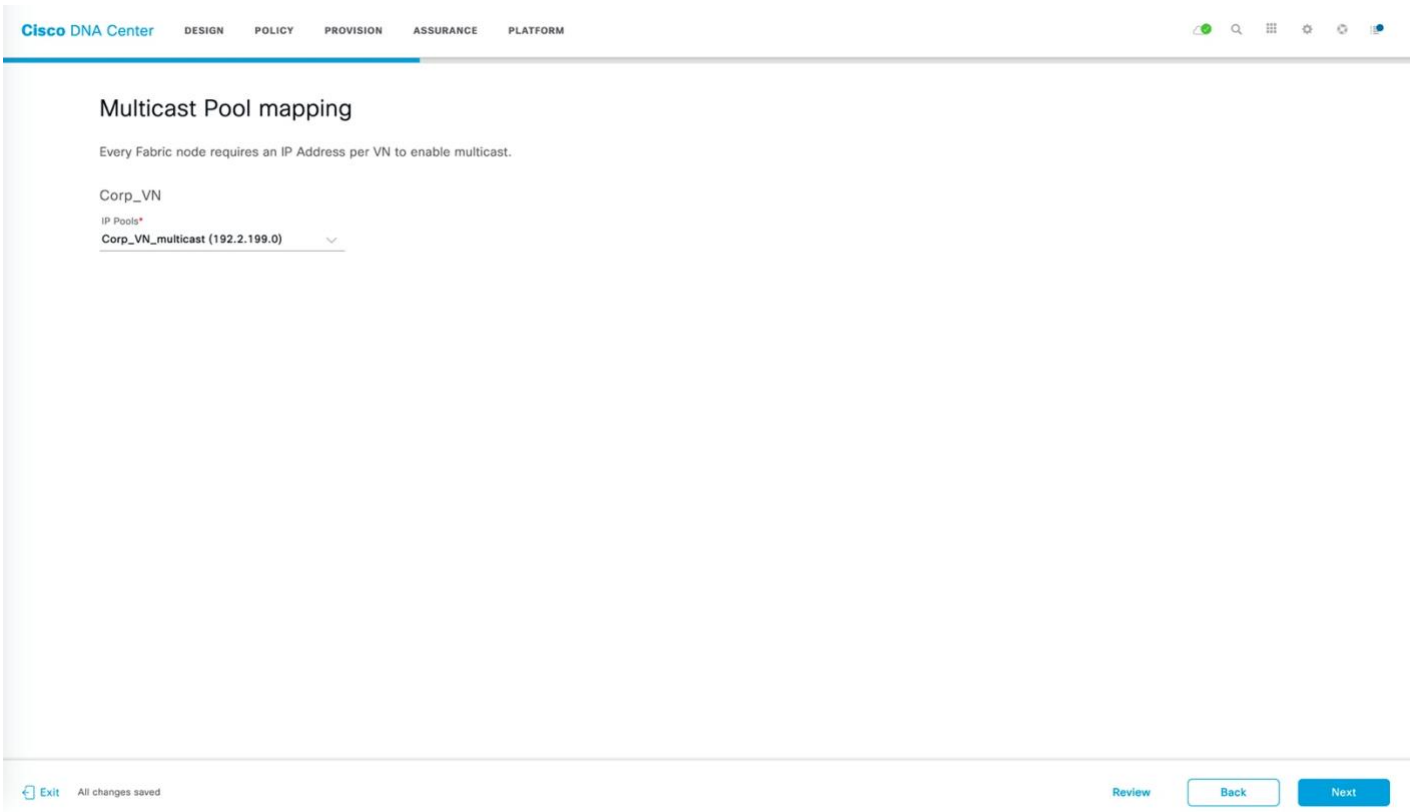


Aninda Chatterjee (aninchat@cisco.com)

Choose what VNs you want to enable this for. In our case, I want this for “Corp_VN”.



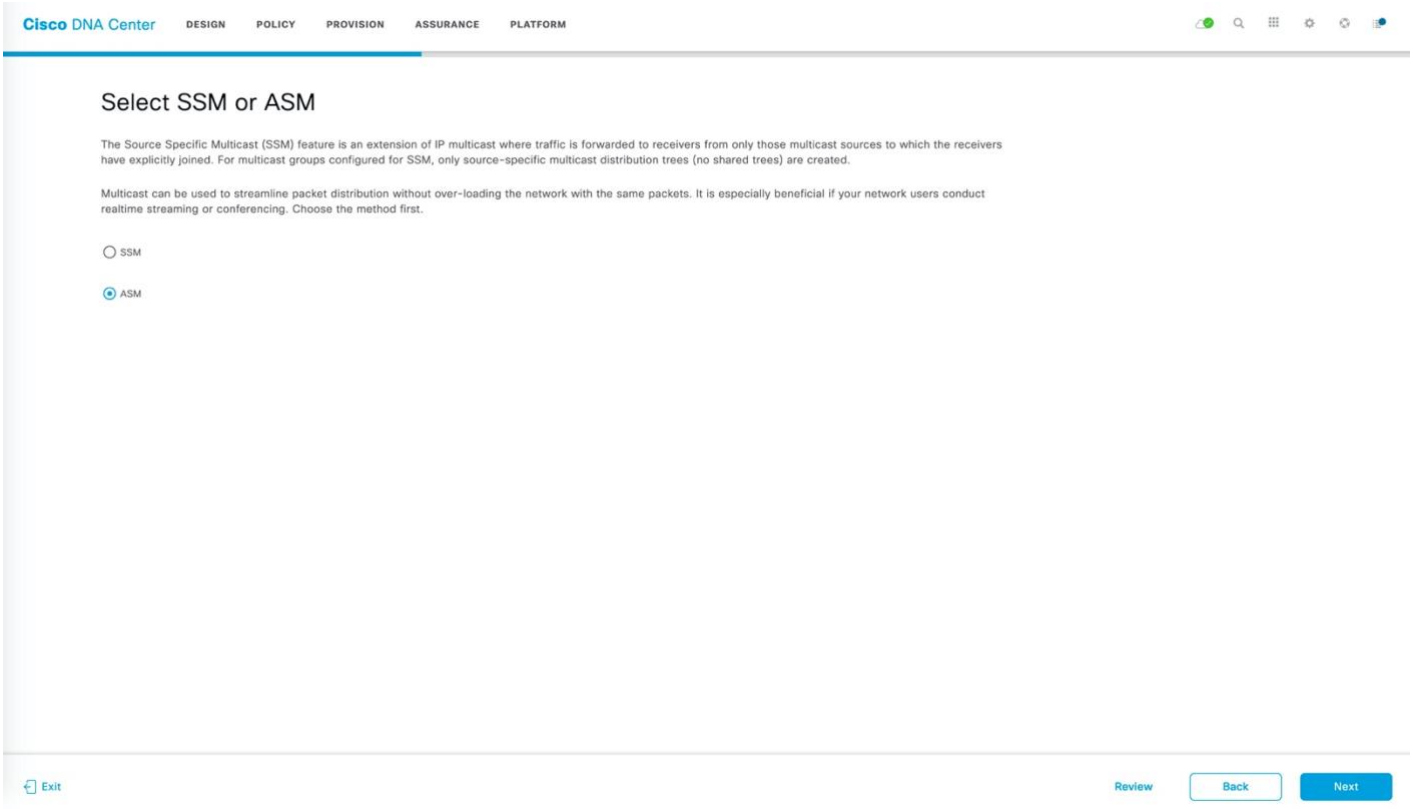
Next, choose a multicast pool:



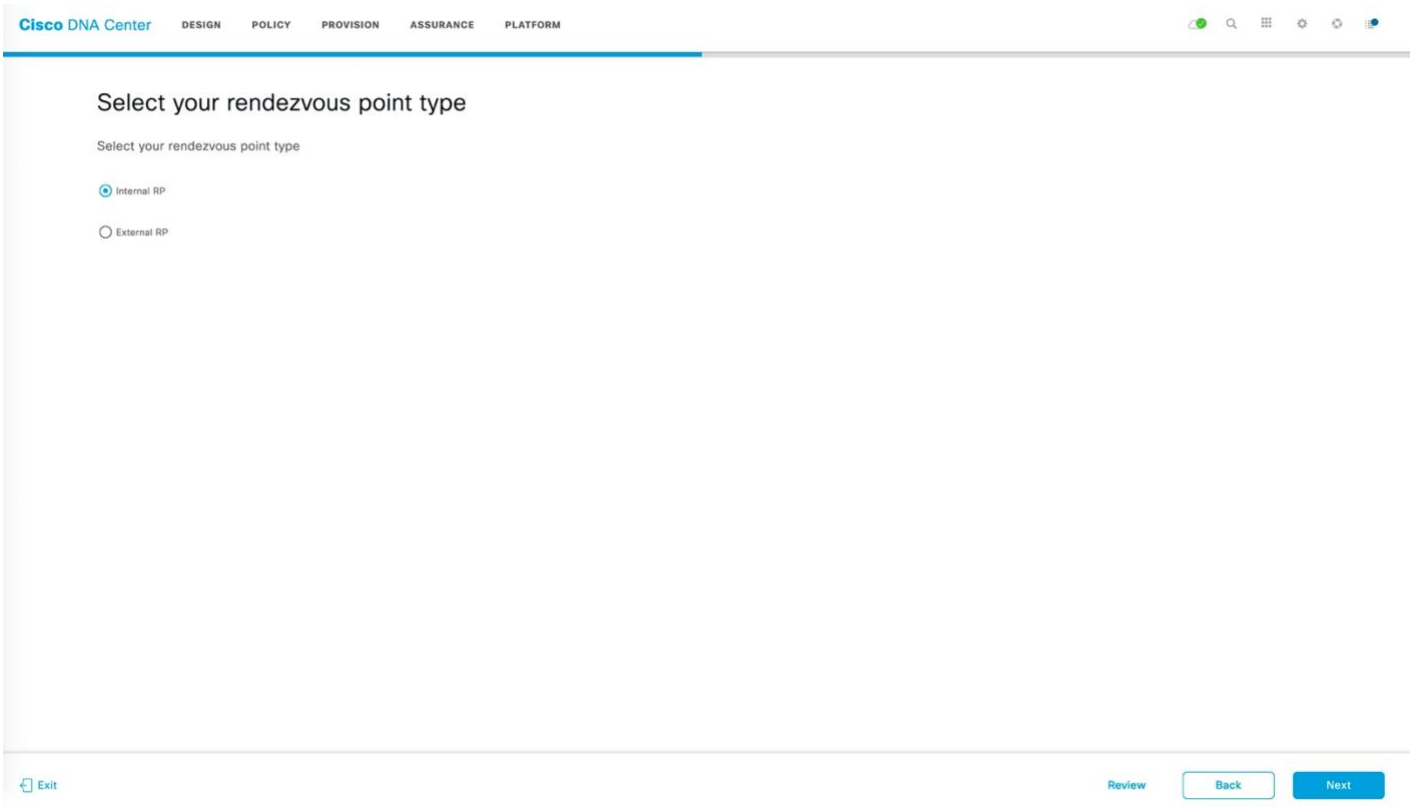
This pool must be pre-defined in DNAC and is used to create loopbacks on the fabric devices within the VRF. We will look at the exact configurations pushed shortly.

Next, choose whether you want to run ASM or SSM in the overlay:

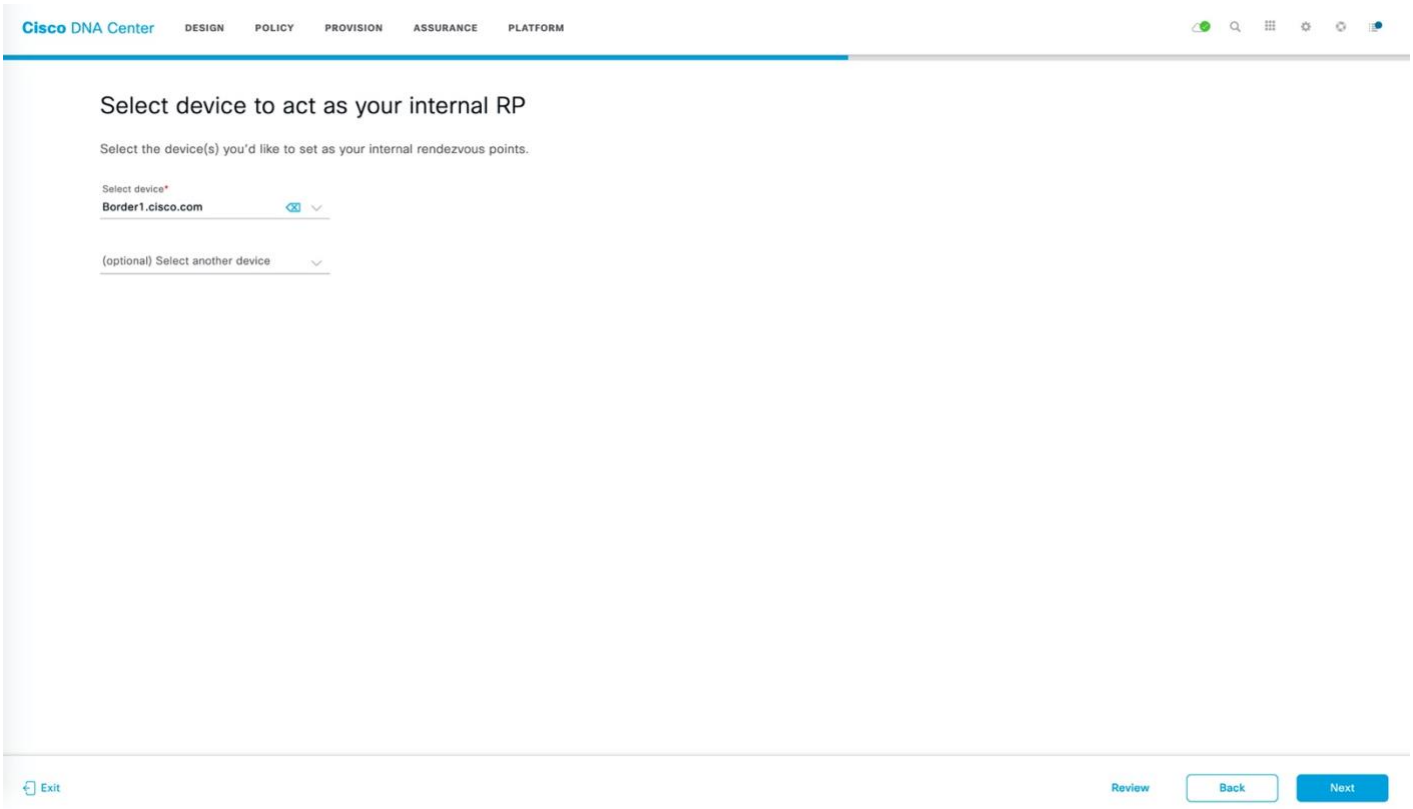
Aninda Chatterjee (aninchat@cisco.com)



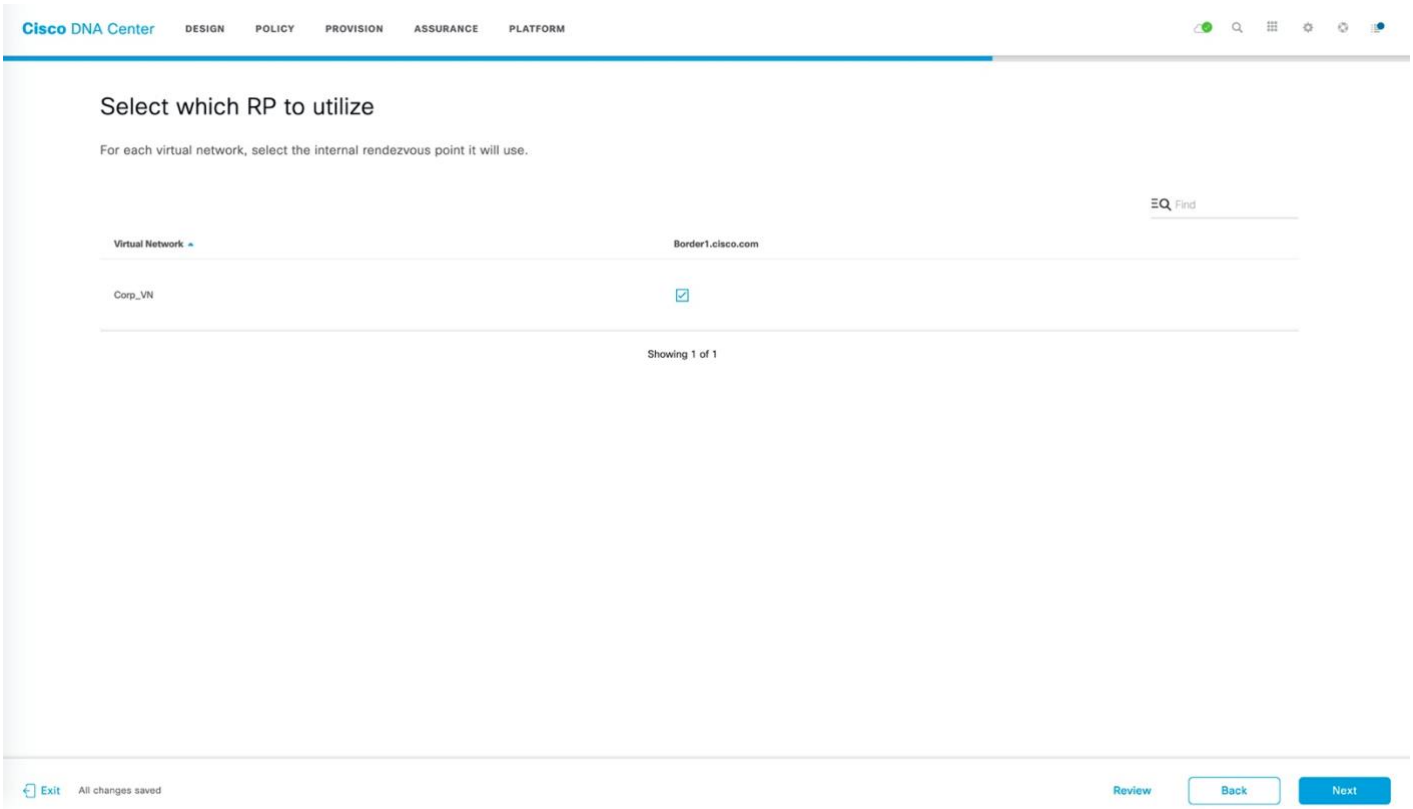
Next, choose where your RP is located – internal (inside the fabric) or external (outside the fabric). In our case, we want Border1 to be the RP, so we will choose internal.



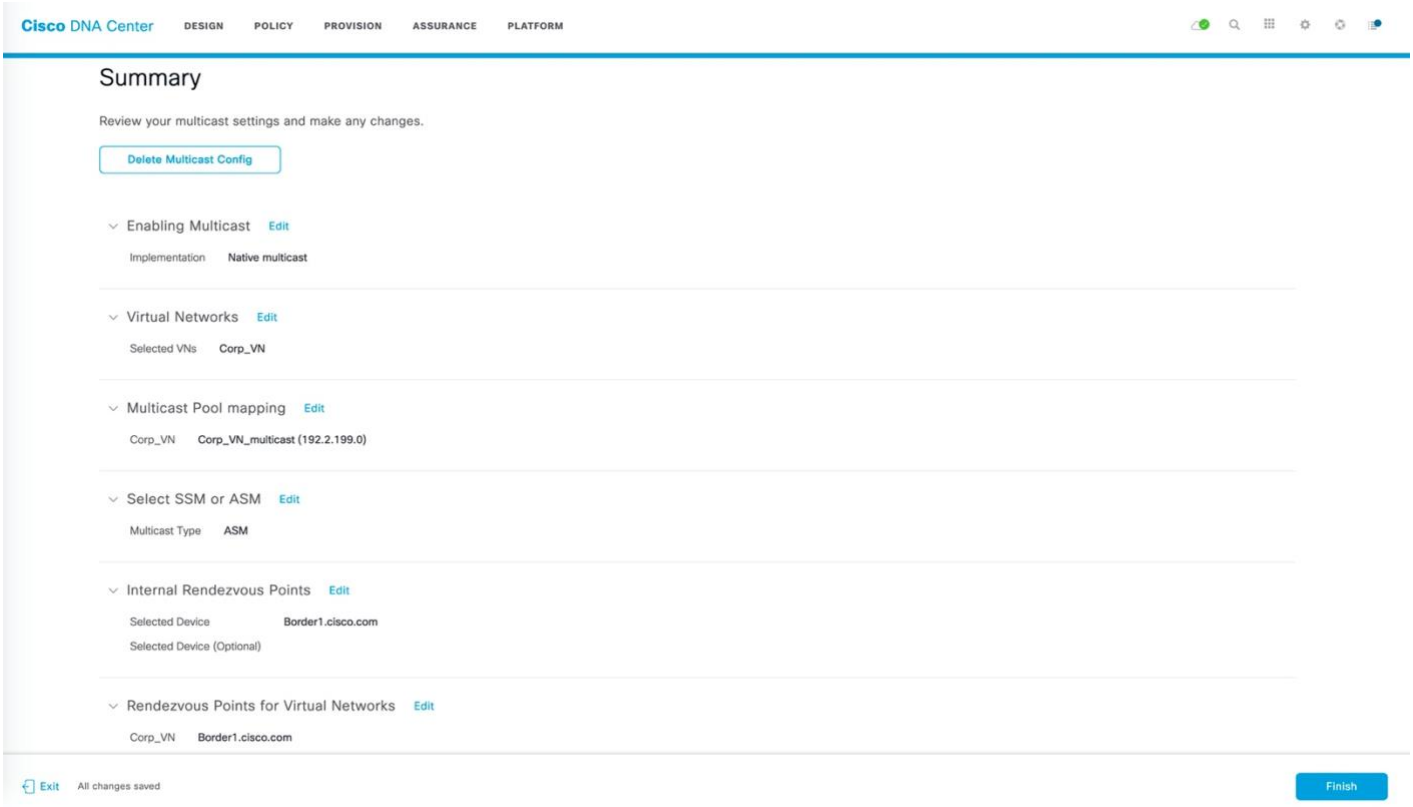
Next, select what device(s) should be the RP. We will select Border1 here.



Next, select the RPs for your VNs.



Once you click ‘Next’, you should now see a summary of all of your choices:



Once you click 'Finish', DNAC will build the appropriate configuration and push them to the devices. In this case, here's what gets pushed:

Border1:

```
// multicast routing enabled for the VRF

Jul 15 2020 12:52:44.626 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast-routing vrf Corp_VN

// transport configured as multicast along with usable SSM range

Jul 15 2020 12:52:44.661 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface LISP0.4100
Jul 15 2020 12:52:44.723 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp transport multicast
Jul 15 2020 12:52:44.780 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp core-group-range 232.0.0.1 1000

// PIM sparse mode enabled under SVI in the VRF

Jul 15 2020 12:52:44.806 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan3014
Jul 15 2020 12:52:44.834 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode
```


Aninda Chatterjee (aninchat@cisco.com)

```
// new loopback created from multicast pool

Jul 15 2020 12:52:44.902 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4100
Jul 15 2020 12:52:44.947 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
Jul 15 2020 12:52:45.015 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.1 255.255.255.255
Jul 15 2020 12:52:45.051 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// RP address specified for the VRF along with source that should be used

Jul 15 2020 12:52:45.118 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN rp-address 192.2.199.1
Jul 15 2020 12:52:45.255 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN register-source Loopback4100

// default SSM range to be used for the VRF

Jul 15 2020 12:52:45.305 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN ssm default
```

Edge1:

```
// enable multicast routing for the VRF

012366: Jul 15 12:53:18.128: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast-routing vrf Corp_VN

// enable multicast transport for LISP and configure allowed SSM range

012367: Jul 15 12:53:18.141: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface LISP0.4100
012368: Jul 15 12:53:18.166: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp transport multicast
012369: Jul 15 12:53:18.185: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp core-group-range 232.0.0.1 1000

// enable IGMPv3 on SVIs that are part of this VRF

012370: Jul 15 12:53:18.199: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan1029
012371: Jul 15 12:53:18.216: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp explicit-tracking
012372: Jul 15 12:53:18.226: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim passive
012373: Jul 15 12:53:18.245: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp version 3
012374: Jul 15 12:53:18.249: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:exit
```


Aninda Chatterjee (aninchat@cisco.com)

```
// create loopback and assign IP from the multicast pool

012385: Jul 15 12:53:18.380: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4100
012386: Jul 15 12:53:18.399: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
012387: Jul 15 12:53:18.418: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.2 255.255.255.255
012388: Jul 15 12:53:18.434: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// configure RP address for the VRF along with source interface

012389: Jul 15 12:53:18.463: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN rp-address 192.2.199.1
012390: Jul 15 12:53:18.492: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN register-source Loopback4100

// default SSM range to be used for the VRF

012391: Jul 15 12:53:18.501: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN ssm default
```

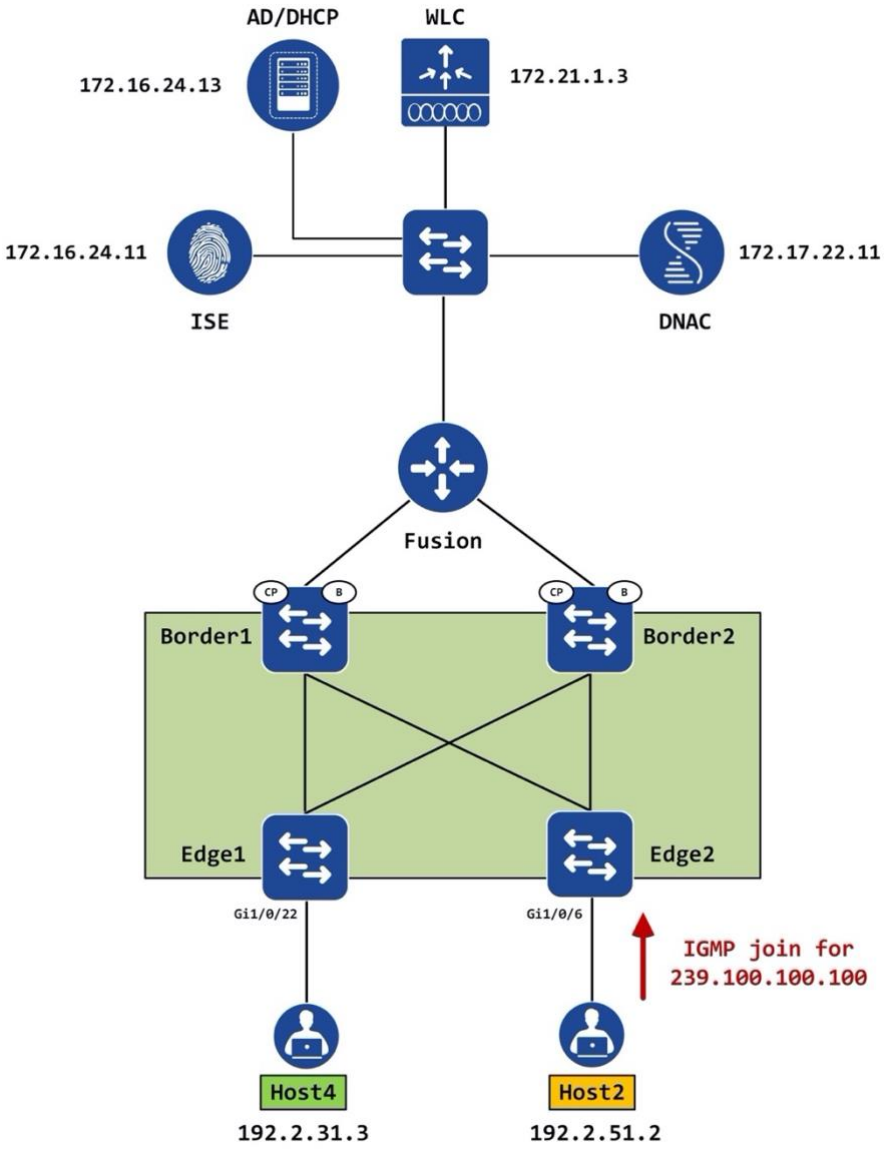
Similar configuration will be pushed to Edge2 as well so I’m not going to show that here.

Aninda Chatterjee (aninchat@cisco.com)

ASM – source and receiver in the fabric

Building the shared tree

Host2 sends an IGMPv3 join (without specifying a source, which essentially makes it an any source join for that group) for 239.100.100.100.



The following debugs (on Edge2) confirm the receipt of the IGMP join:

```
098339: Jun 27 23:05:27.770: IGMP(2): Received v3 Report for 1 group on Vlan1030 from
192.2.51.2
098340: Jun 27 23:05:27.770: IGMP(2): Received Group record for group 239.100.100.100, mode 4
from 192.2.51.2 for 0 sources
098341: Jun 27 23:05:27.770: IGMP(2): WAVL Insert group: 239.100.100.100   interface: Vlan1030
Successful
098343: Jun 27 23:05:27.770: IGMP(2): Switching to EXCLUDE mode for 239.100.100.100 on Vlan1030
098344: Jun 27 23:05:27.770: IGMP(2): Updating EXCLUDE group timer for 239.100.100.100
098345: Jun 27 23:05:27.770: IGMP_ET(2): Adding Host 192.2.51.2 for 239.100.100.100 on Vlan1030
098346: Jun 27 23:05:27.770: IGMP_ET(2): host 192.2.51.2 report for 239.100.100.100 (0 srcs) on
Vlan1030
098347: Jun 27 23:05:27.770: IGMP_ET(2): host 192.2.51.2 switch to exclude for 239.100.100.100
on Vlan1030
098348: Jun 27 23:05:27.770: IGMP(2): MRT Add/Update Vlan1030 for (*,239.100.100.100) by 1
```


Aninda Chatterjee (aninchat@cisco.com)

The group membership can be confirmed as follows:

```
Edge2#show ip igmp vrf Corp_VN groups 239.100.100.100
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter   Group
Accounted
239.100.100.100  Vlan1030                 00:01:33  00:02:18  192.2.51.2
```

```
Edge2#show ip igmp vrf Corp_VN membership 239.100.100.100
Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP, the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude

 Channel/Group                  Reporter        Uptime   Exp.  Flags  Interface
 *,239.100.100.100              0/1             00:01:52 01:59 3AT    Vl1030
```

A (*,G) entry is now created in the multicast routing table (for this VRF):

```
Edge2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:03:28/00:02:24, RP 192.2.199.1, flags: SJC
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65, LISP: [192.2.101.65, 232.0.3.1]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 00:03:28/00:02:24
```

Several important things to dissect here. Remember that for ASM, it is essential to have a Rendezvous Point – both the shared and source trees are rooted at the RP. From the above, we can see that the RP is 192.2.199.1 – this is from configuration that was pushed during multicast deployment via DNAC.

Aninda Chatterjee (aninchat@cisco.com)

The RP can also be confirmed using the following:

```
Edge2#show ip pim vrf Corp_VN rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 192.2.199.1 (?)
```

The incoming interface is determined by a RPF (reverse path forwarding) lookup for the RP IP address. Let's confirm that it matches what we see in the output earlier:

```
Edge2#show ip cef vrf Corp_VN 192.2.199.1 detail
192.2.199.1/32, epoch 1, flags [subtree context, check lisp eligibility]
  SC owned,sourced: LISP remote EID - locator status bits 0x00000001
  LISP remote EID: 68392 packets 1367238 bytes fwd action encap, cfg as EID space
  LISP source path list
    nexthop 192.2.101.65 LISP0.4100
  2 IPL sources [no flags]
  nexthop 192.2.101.65 LISP0.4100
```

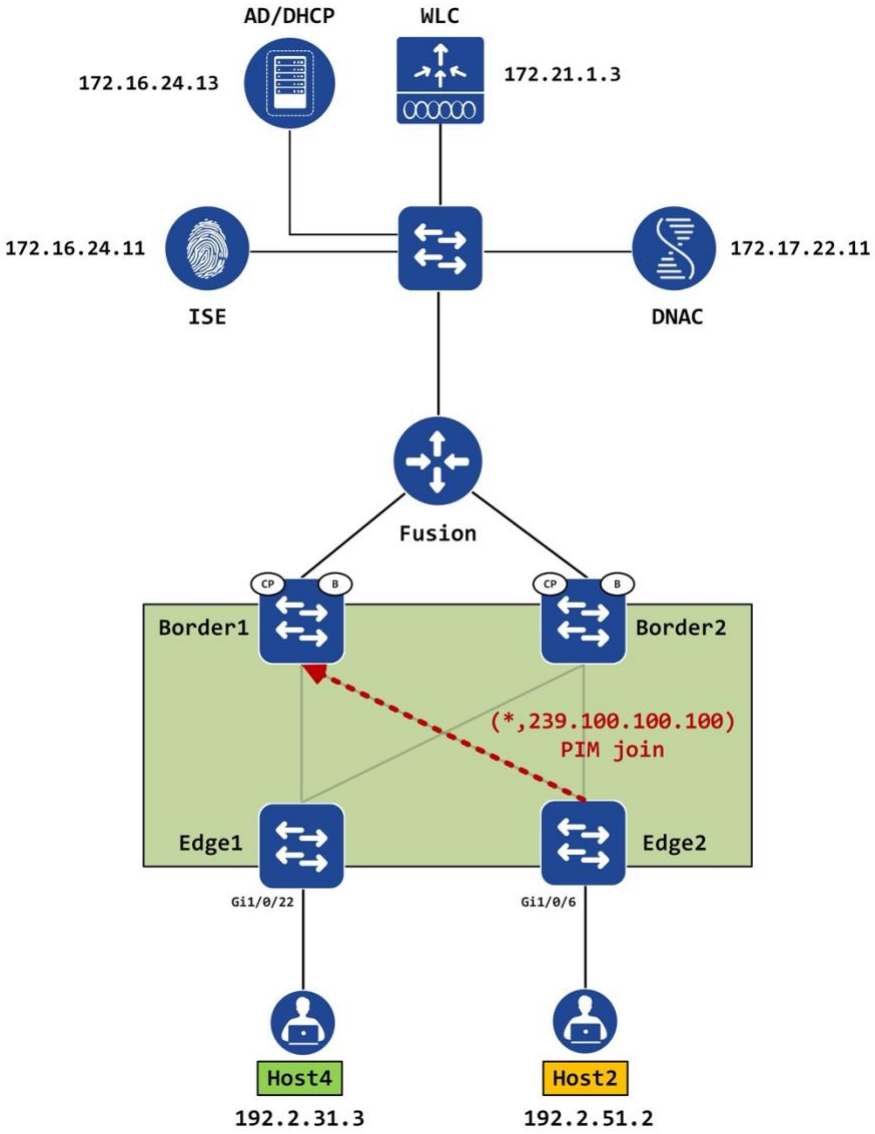
Good – so we know the RPF interface and next-hop is correctly populated in the multicast routing table and is in accordance with what we see in CEF. The outgoing interface in the multicast routing table is the SVI 1030 – this is where the host is connected and the L3 interface that processed the IGMP join.

The last important thing to look at is the new underlay (S,G) that is derived and shown with the *"verbose"* option of the command. After the incoming interface information, you can see that we have another group listed – [192.2.101.65, 232.0.3.1]. 232.0.3.1 is the underlay SSM group that was derived for the overlay group of 239.10.10.10, while 192.2.101.65 is the source (which is the next-hop towards the RP).

We're not going to go into detail of the actual hash algorithm that generates the underlay SSM group – just know that for every ASM group in the overlay, a corresponding SSM group in the underlay is derived and used.

Outside of the actual (*,G) entry creation on Edge2 itself, the IGMP join also triggers a PIM (*,G) join that is sent towards the RP (which is Border1) by Edge2.

Aninda Chatterjee (aninchat@cisco.com)



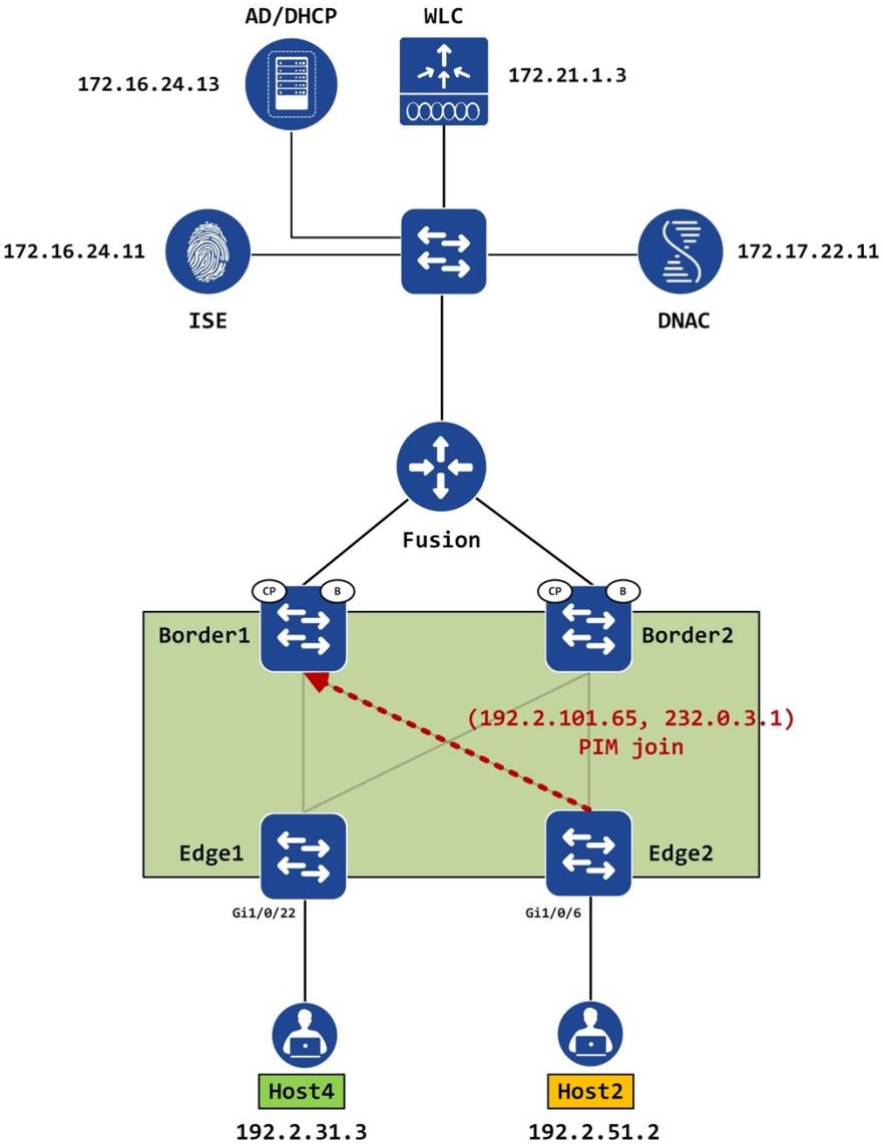
The underlay is purely SSM and we already have a (S,G) derived for it, as seen earlier – (192.2.101.65, 232.0.3.1). A PIM (S,G) join is sent for this in the underlay. This is for the corresponding (*,G) of the overlay which is why the source is the RP itself (the IP address that is used to resolve the RP). Debugs taken on Edge2 show transmission of this:

```
098366: Jun 27 23:05:27.771: PIM(0): Building Join/Prune packet for nbr 192.2.101.72
098367: Jun 27 23:05:27.771: PIM(0):  Adding v2 (192.2.101.65/32, 232.0.3.1), S-bit
Join
098368: Jun 27 23:05:27.771: PIM(0): Send v2 join/prune to 192.2.101.72
(GigabitEthernet1/0/3)
```

Border1 debugs show receipt of this join:

```
Jun 27 23:05:27.777: PIM(0): Received v2 Join/Prune on GigabitEthernet1/0/3 from
192.2.101.73, to us
Jun 27 23:05:27.777: PIM(0): Join-list: (192.2.101.65/32, 232.0.3.1), S-bit set
Jun 27 23:05:27.778: PIM(0): Add GigabitEthernet1/0/3/192.2.101.73 to (192.2.101.65,
232.0.3.1), Forward state, by PIM SG Join
```


Aninda Chatterjee (aninchat@cisco.com)



A packet captures shows a PIM join for both of these:

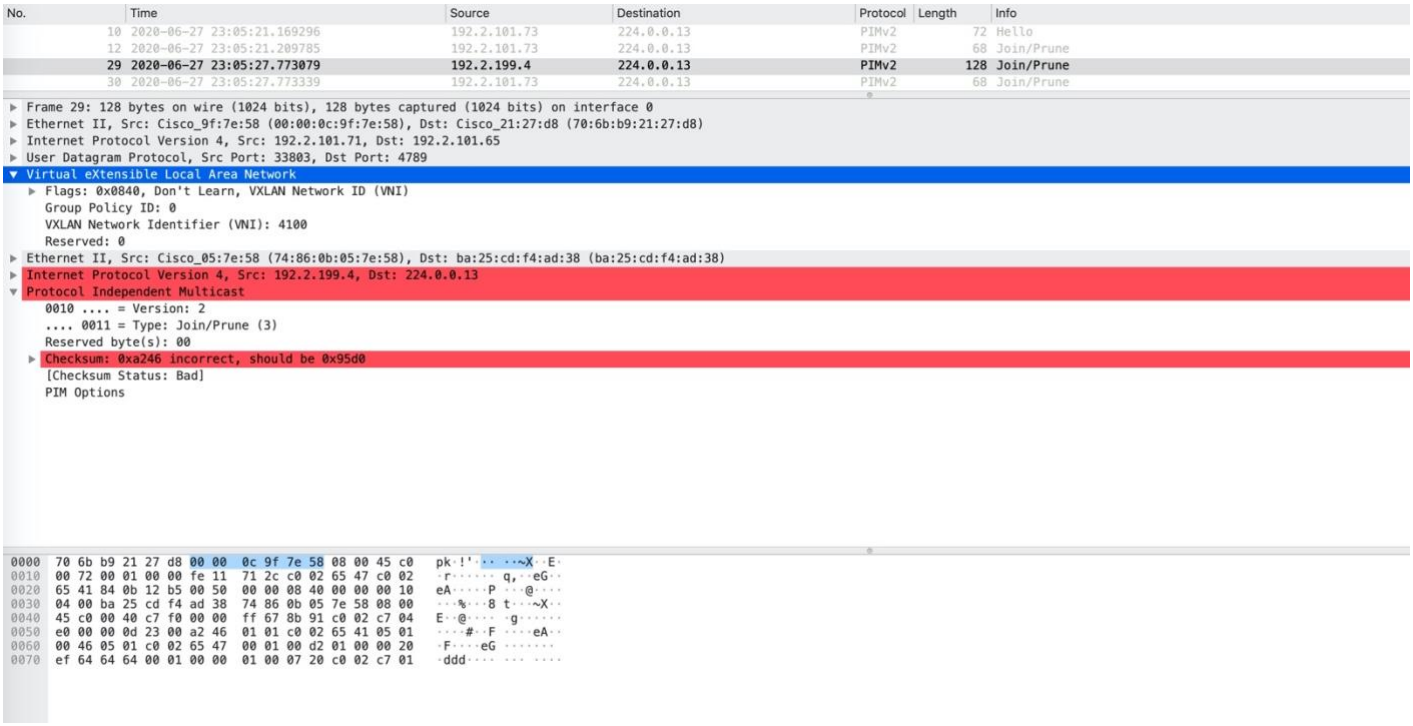
No.	Time	Source	Destination	Protocol	Length	Info
10	2020-06-27 23:05:21.169296	192.2.101.73	224.0.0.13	PIMv2	72	Hello
12	2020-06-27 23:05:21.209785	192.2.101.73	224.0.0.13	PIMv2	68	Join/Prune
29	2020-06-27 23:05:27.773079	192.2.199.4	224.0.0.13	PIMv2	128	Join/Prune
30	2020-06-27 23:05:27.773339	192.2.101.73	224.0.0.13	PIMv2	68	Join/Prune
52	2020-06-27 23:05:35.610780	192.2.199.4	224.0.0.13	PIMv2	128	Join/Prune

```
▶ Frame 30: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▶ Ethernet II, Src: Cisco_05:7e:58 (74:86:0b:05:7e:58), Dst: IPv4mcast_0d (01:00:5e:00:00:0d)
▶ Internet Protocol Version 4, Src: 192.2.101.73, Dst: 224.0.0.13
▼ Protocol Independent Multicast
    0010 .... = Version: 2
    .... 0011 = Type: Join/Prune (3)
    Reserved byte(s): 00
    Checksum: 0x9f5a [correct]
    [Checksum Status: Good]
  ▼ PIM Options
      Upstream-neighbor: 192.2.101.72
      Reserved byte(s): 00
      Num Groups: 1
      Holdtime: 210
    ▼ Group 0: 232.0.3.1/32
      ▼ Num Joins: 1
          IP address: 192.2.101.65/32 (S)
        Num Prunes: 0
```

A source is specified in the join, making it a (S,G) PIM join. The upstream neighbor is also set to the IP address of the next hop to reach 192.2.101.65.

This second packet shows the (*,G) join:

Aninda Chatterjee (aninchat@cisco.com)



This join was sent in the overlay, which is why you see the VXLAN encapsulation as well, specifying the instance ID for this VN (4100). The reason that Wirehshark says this packet has an incorrect checksum is because of the new LISP join attributes – Wireshark does not have dissectors to understand this.

The last line of the hex dump confirms what group this was for – ‘ef 64 64 64’ translates to 239.100.100.100.

Border1 should now have a (*,G) entry for 239.100.100.100 in the overlay (in the corresponding VRF multicast routing table) and it should also have a (S,G) entry for 192.2.101.65 as the source and 232.0.3.1 as the group.

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:04:04/00:03:21, RP 192.2.199.1, flags: Sp
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.65, 232.0.3.1), Forward/Sparse, 00:04:04/stopped, p
      192.2.101.71, 00:04:04/00:03:21
```

```
Border1#show ip mroute 232.0.3.1 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.65, 232.0.3.1), 00:05:39/00:02:45, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/3, Forward/Sparse, 00:05:39/00:02:45, p
```

Edge2 should also have this (S,G) in the underlay:

```
Edge2#show ip mroute 232.0.3.1 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.65, 232.0.3.1), 00:05:18/00:00:41, flags: sT
  Incoming interface: GigabitEthernet1/0/3, RPF nbr 192.2.101.72
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 00:05:18/stopped
```

The shared tree (in the overlay) is rooted at Border1 (since it is the RP), which is why the incoming interface is Null0.

Aninda Chatterjee (aninchat@cisco.com)

The outgoing interface on Edge2 for the SSM entry is Null0 because the packet should be decapsulated here.

The MFIB (multicast FIB) entries give a clearer picture of how the encapsulation/decapsulation would happen on the shared tree. Border1 sees the overlay entry with an outgoing interface list of a LISP interface, with encapsulation information populated from CEF:

```
Border1#show ip mfib vrf Corp_VN 239.100.100.100 verbose
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
VRF Corp_VN
 (*,239.100.100.100) Flags: C K HW
   0xF9  OIF-IC count: 0, OIF-A count: 1
   SW Forwarding: 8/0/60/0, Other: 0/0/0
   HW Forwarding:   0/0/0/0, Other: 0/0/0
   Tunnel3 Flags: RA A MA
   LISP0.4100, (192.2.101.65, 232.0.3.1) Flags: RF F NS
     CEF: Adjacency with MAC:
4500000000004000001 16AA8C0026541E8000301000012B50000000008480000001004 00BA25CDF4AD38BA2
5CDF4AD380000
     Pkts: 0/5
```

C0026541 – 192.2.101.65
E8000301 – 232.0.3.1
12B5 – 4789 (well-known destination port for VXLAN)
001004 – 4100 (VNI in VXLAN header)

The SSM underlay MFIB entry also gives us an understanding of how the Ethernet header will be populated:

```
Border1#show ip mfib 232.0.3.1 verbose
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
Default
 (192.2.101.65,232.0.3.1) Flags: K HW
   0x1A8  OIF-IC count: 0, OIF-A count: 1
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   9/0/116/0, Other: 0/0/0
   Null0 Flags: RA A MA
   GigabitEthernet1/0/3 Flags: RF F NS
     CEF: Adjacency with MAC: 01005E000301706BB92127D80800
     Pkts: 0/0
```

01005E000301 – destination mac in outer Ethernet header
706BB92127D8 – source mac in outer Ethernet header
0800 – Type (0x0800 is IPv4)

Similarly, on Edge2, we see the SSM underlay entry in MFIB:

```
Edge2#show ip mfib 232.0.3.1 verbose
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
Default
 (192.2.101.65,232.0.3.1) Flags: K HW
   0x319  OIF-IC count: 0, OIF-A count: 1
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   9/0/128/0, Other: 0/0/0
   GigabitEthernet1/0/3 Flags: RA A MA
   Null0, LISPv4 Decap Flags: RF F NS
     CEF: OCE (lisp decap)
     Pkts: 0/0
```

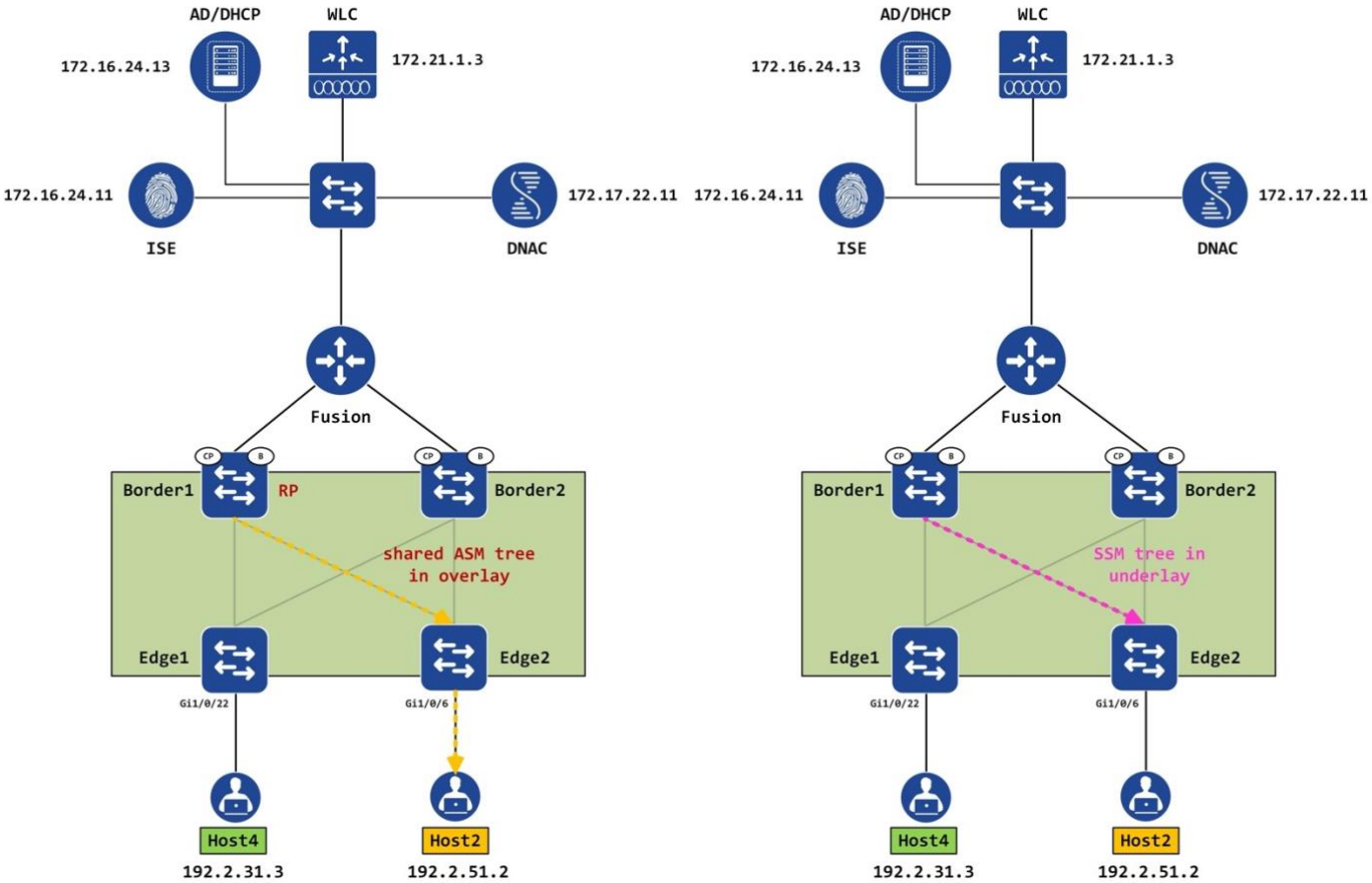

Aninda Chatterjee (aninchat@cisco.com)

As seen from the above, Gi1/0/3 (which connects to Border1) is marked as an ‘Accept’ interface and Null0 is the ‘Forward’ interface, which also points to decapsulation of the packet.

For the overlay, a corresponding entry exists in MFIB as well:

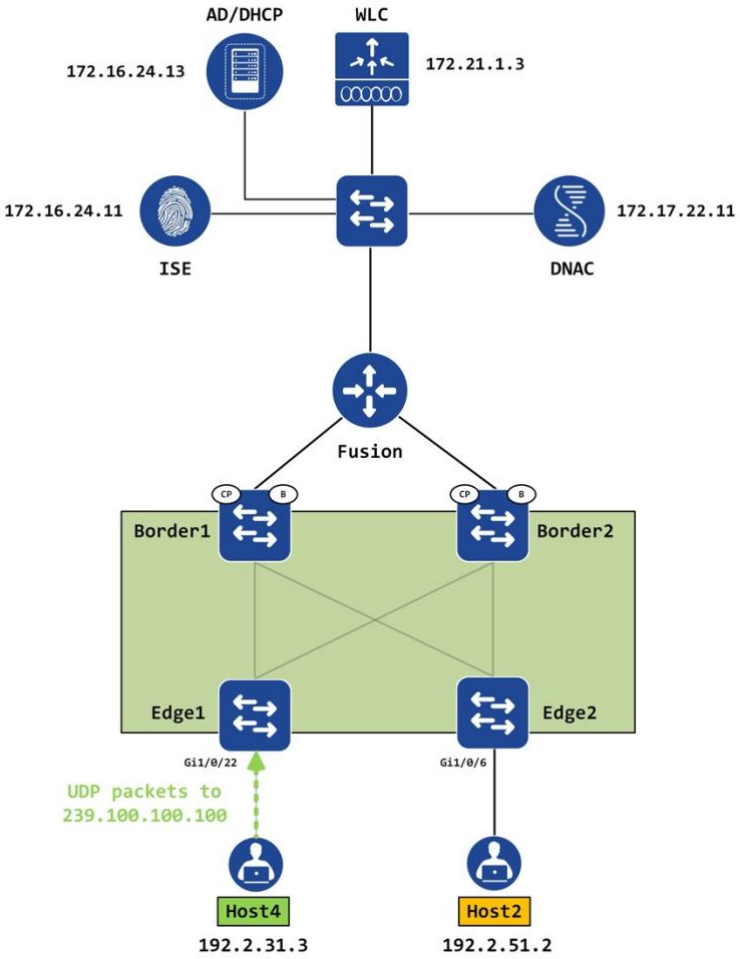
```
Edge2#show ip mfib vrf Corp_VN 239.100.100.100 verbose

*snip*

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
VRF Corp_VN
 (*,239.100.100.100) Flags: C K HW
   0x39A  OIF-IC count: 0, OIF-A count: 1
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   13/0/86/0, Other: 0/0/0
   LISP0.4100 Flags: RA A MA NS
   Vlan1030 Flags: RF F NS
     CEF: Adjacency with MAC: 01005E64646400000C9FF4650800
     Pkts: 0/0
```

So, now, we have the following two trees built when the receiver has joined:



Start of source stream

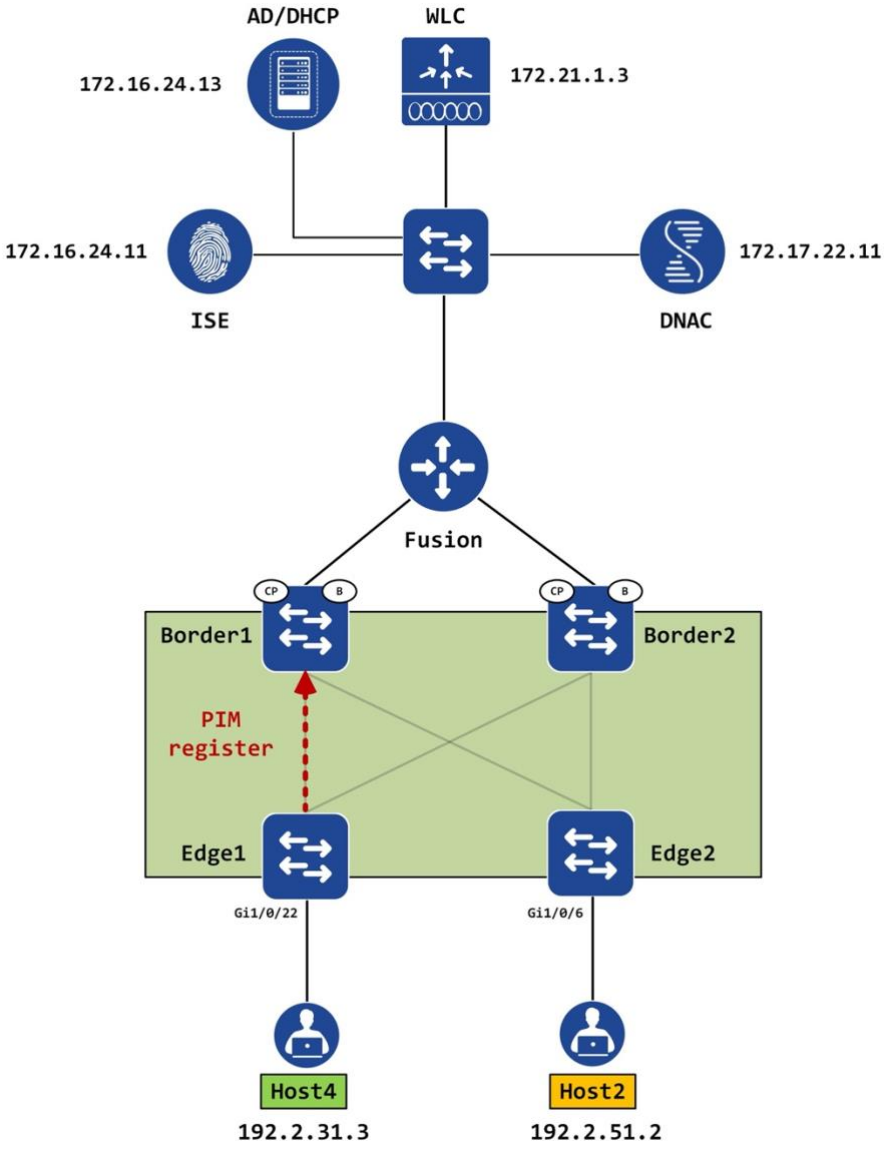
The source, Host4, now starts to send UDP traffic to 239.100.100.100. When this hits Edge1, because of a miss in MFIB, the packet is punted to the CPU to create state in the multicast routing table.




```
// no entry found in mfib, packet is punted to form state

452270: Jul  8 18:06:36.179: FMANRP-MCAST: (192.2.31.3, 239.100.100.100) mfib entry not
found
452271: Jul  8 18:06:36.179: FMANRP-MCAST: (*, 239.100.100.100) mfib entry not found
452272: Jul  8 18:06:36.179: FMANRP-MCAST: Punted packet from Vlan1029 tagged.
452273: Jul  8 18:06:36.186: FMANRP-MCAST(2): add (192.2.31.3, 239.100.100.100/64), id
0x0
452275: Jul  8 18:06:36.186: FM_MCAST_T(2): route_add:ADD(239.100.100.100, 192.2.31.3)
OBJ ID:0x478/0x478
452276: Jul  8 18:06:36.186: FMANRP-MCAST(2): modify (192.2.31.3, 239.100.100.100/64),
id 0x478
452277: Jul  8 18:06:36.186: FMANRP_MCAST(2): add OCE for Vlan1029, flags 0x0 -> 0x10

// mrib adds entry

452283: Jul  8 18:06:36.187: MRIB: Table = VRF Corp_VN, [4  ] i/o - received 36 bytes
452284: Jul  8 18:06:36.187: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Received   DDE* [Vl1029  SP* ]
452285: Jul  8 18:06:36.187: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Updated   DDE* #A=0, #F=0 [Vl1029  SP* ]
452286: Jul  8 18:06:36.187: MRIB: Table = VRF Corp_VN, [4  ]
(192.2.31.3,239.100.100.100) Added
```


Aninda Chatterjee (aninchat@cisco.com)

'DDE' in the above debugs stands for 'Data Driven Event', which essentially means the state creation was due to the data packet that was punted to the CPU. Since Edge1 is the first hop router, it must register this source to the RP (Border1). It does so by sending a unicast encapsulating the actual multicast data and sending it to the RP. During this time, the 'Registering' flag is set as well.



The following debugs on Edge1 show this as well:

```
452341: Jul  8 18:06:36.190: PIM(2): Adding register encap tunnel (Tunnel2) as
forwarding interface of (192.2.31.3, 239.100.100.100).
452342: Jul  8 18:06:36.190: MRT(2): Set the F-flag for (*, 239.100.100.100)
452343: Jul  8 18:06:36.190: MRT(2): Set the F-flag for (192.2.31.3, 239.100.100.100)
452344: Jul  8 18:06:36.190: MRT(2): Start 0x7FAB21D0F1C0 [1251] send join timer
(192.2.31.3, 239.100.100.100) delay 58 sec
452345: Jul  8 18:06:36.190: MRT(2): Create (192.2.31.3,239.100.100.100), RPF
(Vlan1029, 0.0.0.0, 10/1)
452351: Jul  8 18:06:36.191: MRT(2): Set the T-flag for (192.2.31.3, 239.100.100.100)
```


Aninda Chatterjee (aninchat@cisco.com)

The MRIB output during the register process, with the register flag set:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:00:00/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65, LISP: [192.2.101.65, 232.0.3.1]
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 00:00:00/00:02:59, flags: FTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.1), Forward/Sparse, 00:00:00/stopped, p
      192.2.101.65, 00:00:00/00:03:29
```

Notice that the outgoing interface list contains Border1s IP address. The incoming interface points to VLAN1029, which is where the source is connected.

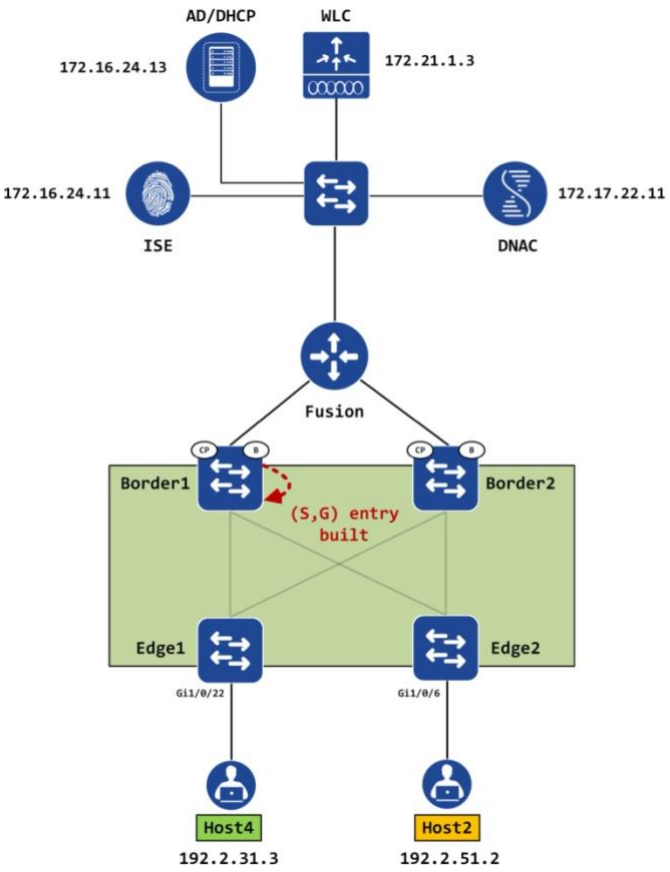
The register packet does two things on the RP (Border1): first, the packet is decapsulated and sent towards the receiver using the shared tree that is already built. Second, it also builds (S,G) state on the RP (Border1) which leads to (S,G) PIM joins sent towards the source S, hop by hop.

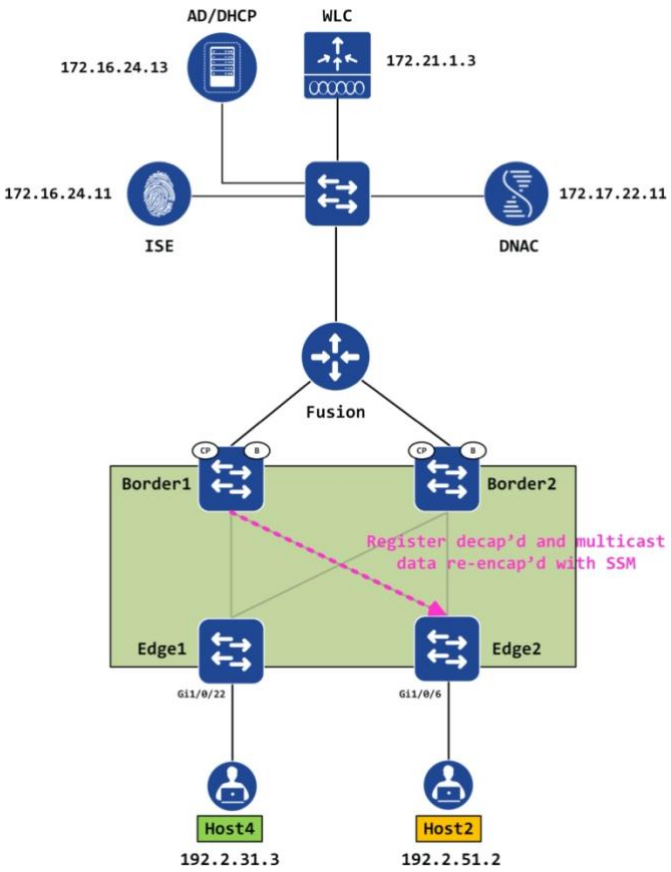
```
// Border1 receives register

Jul  8 18:06:36.685: PIM(117): Received v2 Register on LISP0.4100 from 192.2.199.3
Jul  8 18:06:36.685:      for 192.2.31.3, group 239.100.100.100

// (S,G) state built from register

Jul  8 18:06:36.693: MRT(117): Create (192.2.31.3,239.100.100.100), RPF (unknown,
0.0.0.0, 0/0)
Jul  8 18:06:36.695: MRT(117): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 232.0.3.1 Outer-source: 192.2.101.65 Successful
Jul  8 18:06:36.698: MRT(117): Add LISP0.4100/232.0.3.1/192.2.101.65to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
Jul  8 18:06:36.698: MRT(117): Inherited explicit tracking node for (192.2.31.3,
239.100.100.100) LISP0.4100, (192.2.101.65, 239.100.100.100), rcv: 192.2.101.71
Jul  8 18:06:36.730: MRT(117): (192.2.31.3,239.100.100.100), RPF change from  /0.0.0.0
to LISP0.4100/192.2.101.70
```


Aninda Chatterjee (aninchat@cisco.com)

Border1 now has the following state in its multicast routing table:

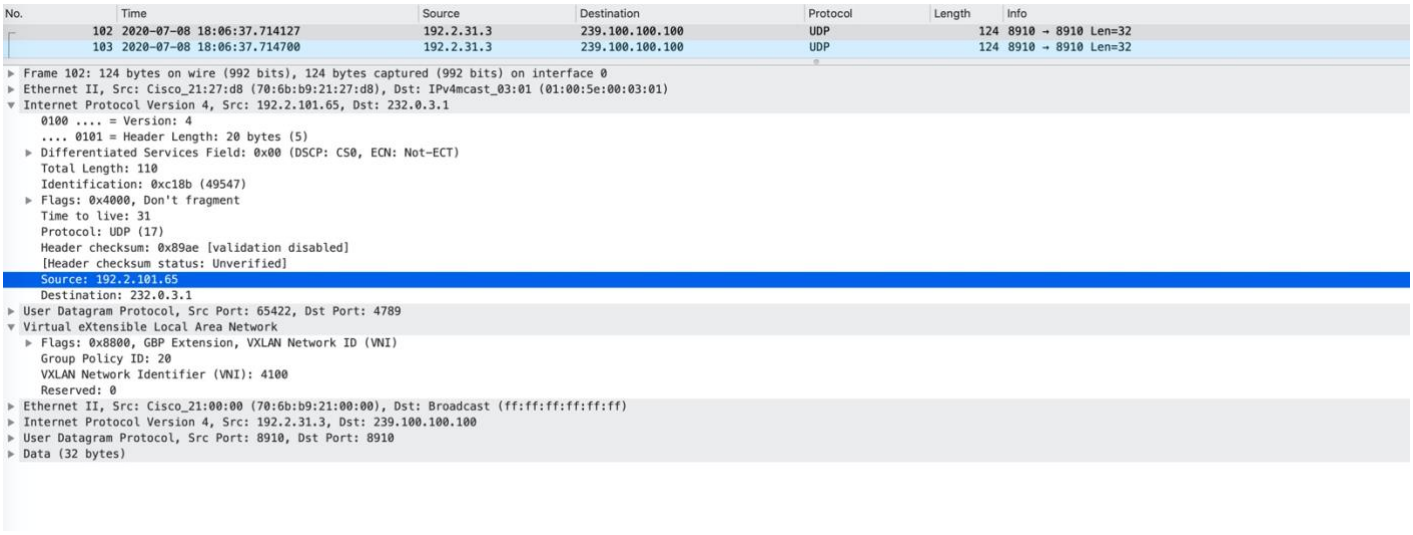
```
Border1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 1w4d/stopped, RP 192.2.199.1, flags: Sp
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.65, 232.0.3.1), Forward/Sparse, 18:22:38/stopped, p
      192.2.101.71, 18:22:38/00:03:16

(192.2.31.3, 239.100.100.100), 00:00:00/00:02:59, flags:
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70, 232.0.3.1]
  Outgoing interface list:
    LISP0.4100, (192.2.101.65, 232.0.3.1), Forward/Sparse, 00:00:00/stopped
      192.2.101.71, 00:00:00/00:03:29
```



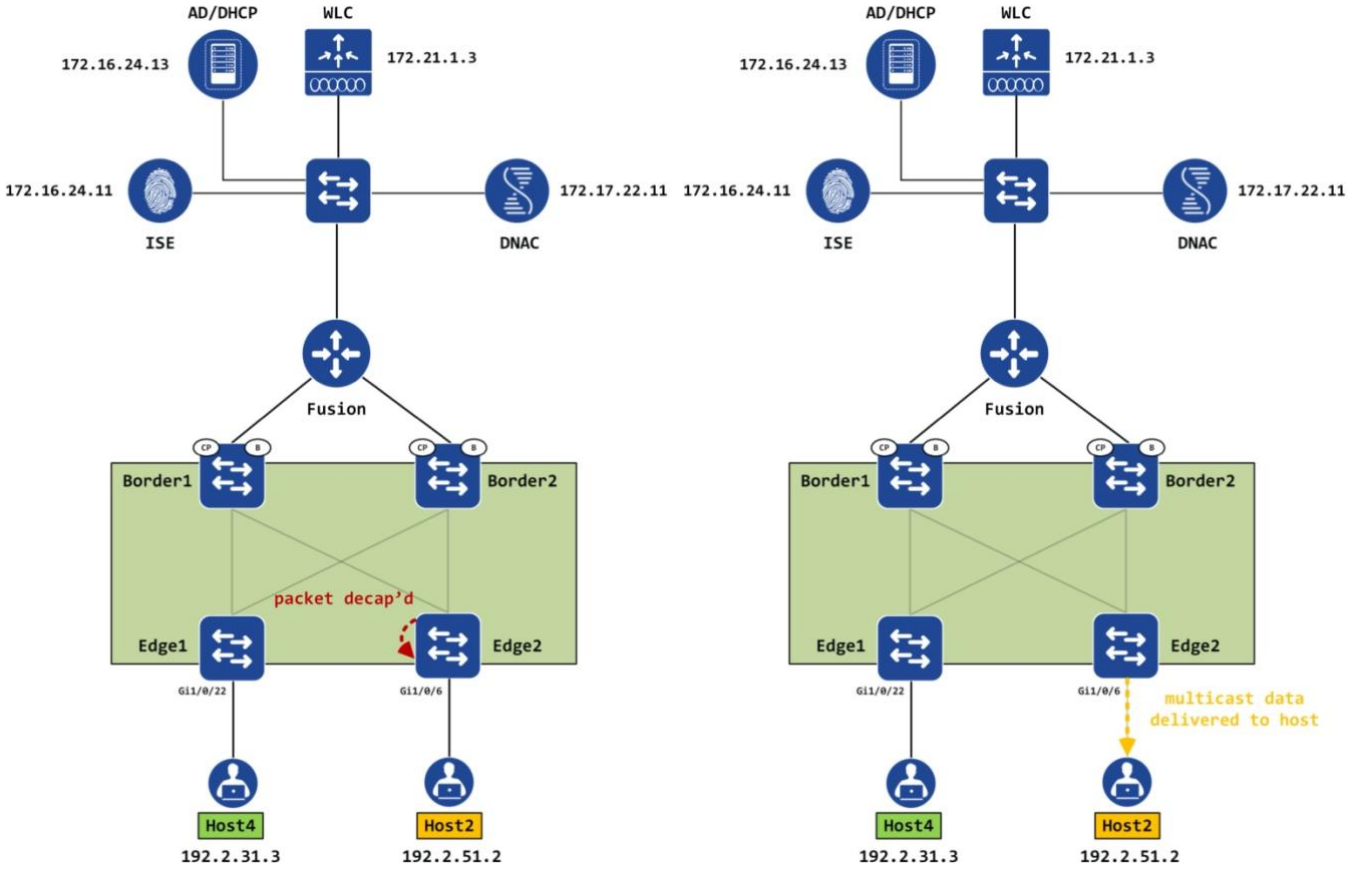


It is important to note here that the multicast data inside the register is not sent as a unicast to Edge2. The underlay SSM is used and the outer header includes the underlay SSM IP address that was derived for the overlay multicast IP address (revisit the section titled ‘Building the shared tree’ to review this again if needed).

A packet capture taken between Border1 and Edge2 proves this:



Notice the source and the destination in the outer IP header – this matches the underlay SSM tree that was built. This packet reaches Edge2, where it is decapsulated and the multicast data is delivered to the host with a second lookup on the decapsulated packet, which hits the overlay multicast routing table for the corresponding VRF.

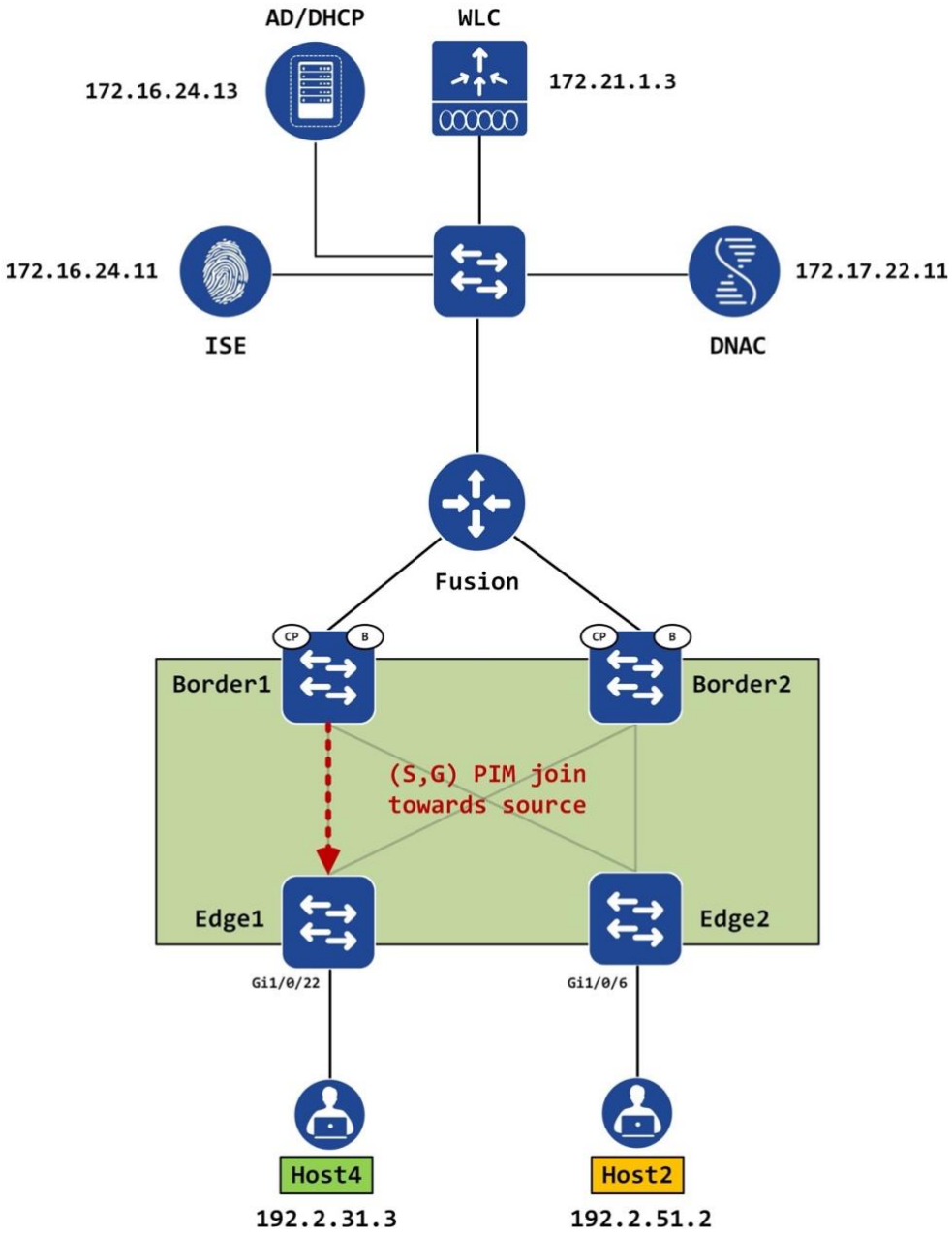


Aninda Chatterjee (aninchat@cisco.com)

Building the source tree

Once the (S,G) state is built on the RP (Border1), it will send a PIM (S,G) join towards the source S. In our case, a (192.2.31.3, 239.100.100.100) PIM join should be sent towards 192.2.31.3.

```
// (S,G) PIM join sent to Edge1

Jul  8 18:06:36.734: PIM(117): Insert (192.2.31.3,239.100.100.100) join in nbr
192.2.101.70's queue
Jul  8 18:06:36.735: PIM(117): Building Join/Prune packet for nbr 192.2.101.70
Jul  8 18:06:36.735: PIM(117):  Adding v2 (192.2.31.3/32, 239.100.100.100), S-bit Join
Jul  8 18:06:36.735: PIM(117): Adding LISP Multicast transport attribute in join/prune
to 192.2.101.70 (LISP0.4100)
Jul  8 18:06:36.735: PIM(117): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```

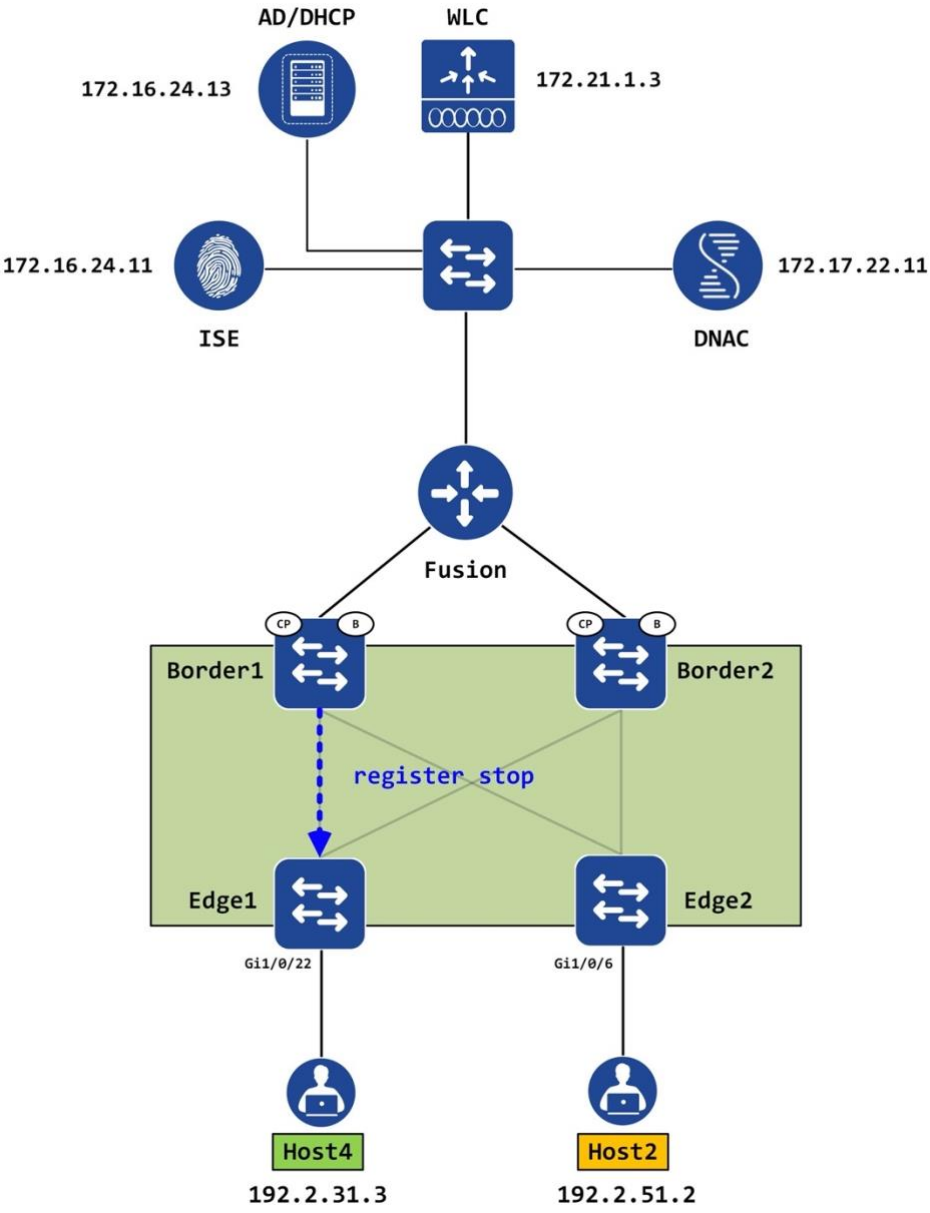


Aninda Chatterjee (aninchat@cisco.com)

This join is received by Edge1 and the existing (S,G) entry is updated and the underlay SSM entry is correctly built as well.

```
// (S,G) state updated on Edge1 after receiving PIM join from Border1

452480: Jul  8 18:06:36.738: PIM(2): J/P Transport Attribute, Transport Type: Multicast
452481: Jul  8 18:06:36.738: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
452482: Jul  8 18:06:36.738: [XTR] LISP-0: Dyn EID mcast join IID 4100 192.2.31.3
192.2.101.65 Created.
452483: Jul  8 18:06:36.738: MRT(2): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 232.0.3.1 Outer-source: 192.2.101.70 Successful
452490: Jul  8 18:06:36.738: MRT(2): Add LISP0.4100/232.0.3.1/192.2.101.70to the olist
of (192.2.31.3, 239.100.100.100), Forward state - MAC not built
452491: Jul  8 18:06:36.738: PIM(2): Add LISP0.4100/192.2.101.65 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join
```

Once the (S,G) is built on Edge1, it starts to natively forward the multicast stream as well (along with the registers). On receiving multicast natively, the RP knows that the source tree has been built and it can now send a register stop to Edge1.



```
// Border1 sends register stop

Jul  8 18:06:37.715: PIM(117): Received v2 Register on LISP0.4100 from 192.2.199.3
Jul  8 18:06:37.715:      for 192.2.31.3, group 239.100.100.100
Jul  8 18:06:37.715: MRT(117): Start 0xFFA0AB7EF0 [364] expire timer (192.2.31.3,
239.100.100.100) delay 180 sec
Jul  8 18:06:37.716: PIM(117): Send v2 Register-Stop to 192.2.199.3 for 192.2.31.3,
group 239.100.100.100
```

Once Edge1 receives the register stop, it can now stop sending register messages to the RP, clear the register flag and only forward the multicast stream natively, which in the case of SDA, will use the underlay SSM tree.

```
// Edge1 receives register stop

452554: Jul  8 18:06:37.718: PIM(2): Received v2 Register-Stop on LISP0.4100 from
192.2.199.1
452555: Jul  8 18:06:37.718: PIM(2):   for source 192.2.31.3, group 239.100.100.100
452558: Jul  8 18:06:37.718: PIM(2): Removing register encap tunnel (Tunnel2) as
forwarding interface of (192.2.31.3, 239.100.100.100).
452560: Jul  8 18:06:37.718: PIM(2): Clear Registering flag to 192.2.199.1 for
(192.2.31.3/32, 239.100.100.100)
452561: Jul  8 18:06:37.718: MRT(2): Start 0x7FAB21D0F2E0 [2480] data register suppress
timer (192.2.31.3, 239.100.100.100) delay 180 sec
452562: Jul  8 18:06:37.718: MRT(2): Start 0x7FAB21D0F5B0 [1880] header register
suppress timer (192.2.31.3, 239.100.100.100) delay 120 sec
452563: Jul  8 18:06:37.718: MRT(2): Start 0x7FAB21D0F5E0 [2480] expire timer
(192.2.31.3, 239.100.100.100) delay 180 sec
```


Aninda Chatterjee (aninchat@cisco.com)

SPT cutover

At the same time, the packet that was decapsulated is also sent towards the receiver over the shared tree. Edge2 receives this and creates (S,G) state in its MRIB after a MFIB miss. This is where things get really interesting.

```
// MFIB miss creates (S,G) state

112111: Jul  8 18:06:37.714: FMANRP-MCAST: (192.2.31.3, 239.100.100.100) mfib entry not
found
112112: Jul  8 18:06:37.714: FMANRP-MCAST: (*, 239.100.100.100) mfib entry not found
112113: Jul  8 18:06:37.714: FMANRP-MCAST: (192.2.31.3, 239.100.100.100) mfib entry not
found
```

```
// (S,G) state created in multicast routing table

112159: Jul  8 18:06:37.725: MRT(2): Create (192.2.31.3,239.100.100.100), RPF (unknown,
0.0.0.0, 0/0)
112164: Jul  8 18:06:37.725: MRT(2): WAVL Insert interface: Vlan1030 in
(192.2.31.3,239.100.100.100) Successful
112165: Jul  8 18:06:37.725: MRT(2): set min mtu for (192.2.31.3, 239.100.100.100)
18010->9100
112166: Jul  8 18:06:37.725: MRT(2): Start 0x7F8195C83128 [377] expire timer interface:
Vlan1030 (192.2.31.3, 239.100.100.100) delay 180 sec
112174: Jul  8 18:06:37.726: MRT(2): Add Vlan1030/239.100.100.100 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
112175: Jul  8 18:06:37.726: MRT(2): Set the J-flag for (192.2.31.3, 239.100.100.100)
```

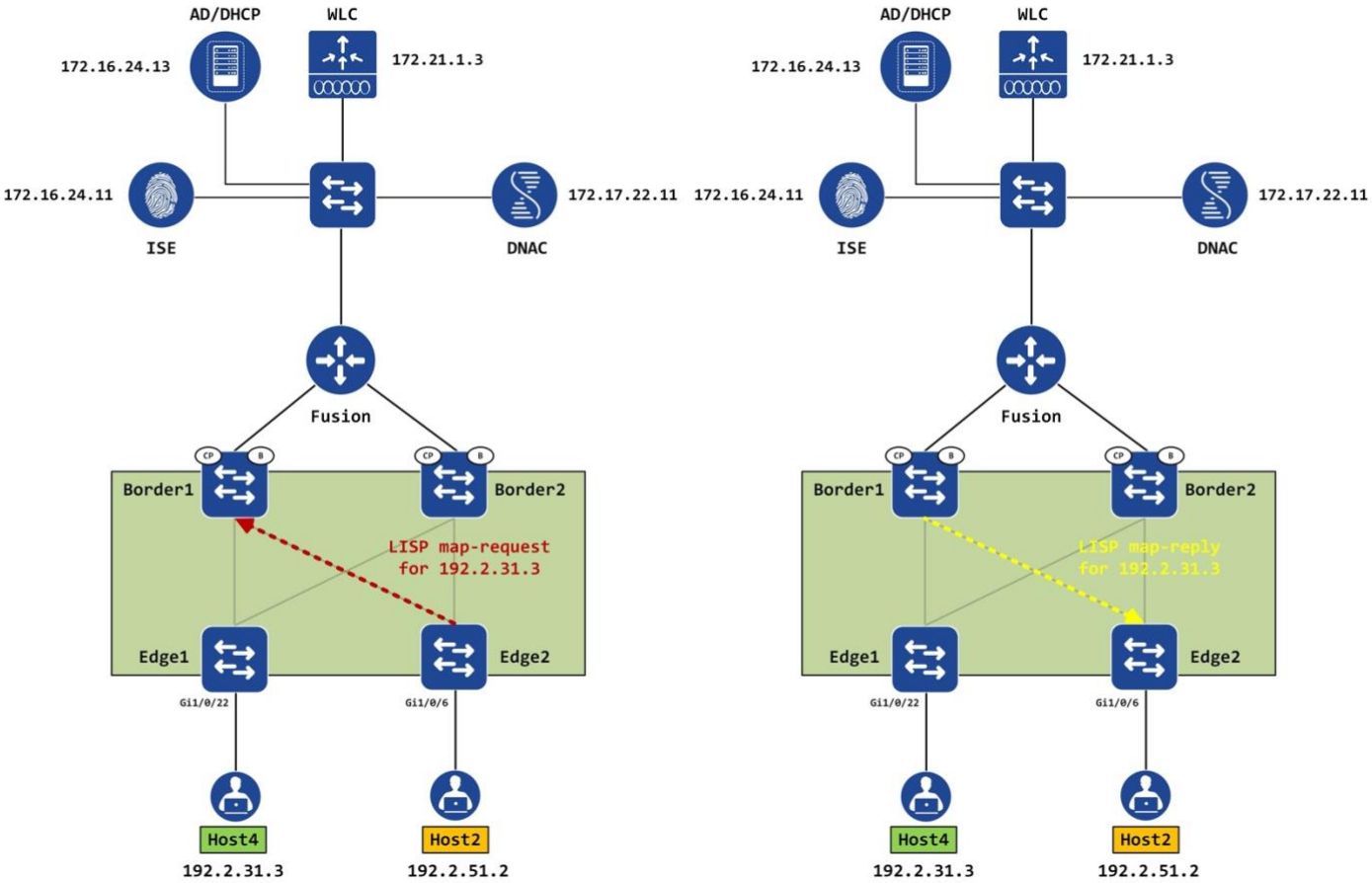
After this, a process called route watch is triggered for the source in order to determine where the source is located.

```
// rwatch triggered for source, along with RPF track for the source

112145: Jul  8 18:06:37.724: MCAST-RWATCH: -af IPv4 -topo 8002: Route watch topology
found.
112146: Jul  8 18:06:37.724: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix not found.
112147: Jul  8 18:06:37.724: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix created.
112148: Jul  8 18:06:37.724: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in RIB.
112149: Jul  8 18:06:37.724: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in LISP.
112150: Jul  8 18:06:37.724: MRT(2): RPF Track start on 192.2.31.3 for (192.2.31.3,
239.100.100.100)
```

This route watch process triggers a resolution of the source via LISP. This leads to the entire process of sending out a map-request for 192.2.31.3, processing a map-reply, updating the LISP map-cache and eventually pushing this information in RIB/FIB.

Aninda Chatterjee (aninchat@cisco.com)

```
// triggered resolution of source location via LISP

112232: Jul  8 18:06:37.774: MRT(2): Processing flags_update check (192.2.31.3,
239.100.100.100)
112233: Jul  8 18:06:37.774: MRIB: Table = VRF Corp_VN, Trans: (192.2.31.3,
239.100.100.100) mroute update: MOD FLAGS, rpf = Null, flag = 180
112234: Jul  8 18:06:37.858:       LISP-0: IID 4100 Request processing of remote EID
prefix map requests to IPv4.
```




```
// send LISP map request

112235: Jul  8 18:06:37.858: [XTR] LISP: Send map request type remote EID prefix
112236: Jul  8 18:06:37.858: [XTR] LISP: Send map request for EID prefix IID 4100
192.2.31.3/32
112237: Jul  8 18:06:37.858: [XTR] LISP-0: Remote EID IID 4100 prefix 192.2.31.3/32,
Send map request (1) (sources: <watch>, state: incomplete, rlocs: 0).
112238: Jul  8 18:06:37.858:       LISP-0: EID-AF IPv4, Sending map-request from
192.2.31.3 to 192.2.31.3 for EID 192.2.31.3/32, ITR-RLOCs 1, nonce 0x7F3F06C8-
0xD2FDBBAD (encap src 192.2.101.71, dst 192.2.101.65), FromPITR.
```


Aninda Chatterjee (aninchat@cisco.com)

```
// process LISP map reply

112241: Jul  8 18:06:37.862: [XTR] LISP: Processing received Map-Reply(2) message on
GigabitEthernet1/0/3 from 192.2.101.65:4342 to 192.2.101.71:4342
112242: Jul  8 18:06:37.862: [XTR] LISP: Received map reply nonce 0x7F3F06C8-
0xD2FDBBAD, records 1
112243: Jul  8 18:06:37.862: [XTR] LISP: Processing Map-Reply mapping record for IID
4100 SVC_IP_IAF_IPv4 192.2.31.3/32 LCAF 2, ttl 1440, action none, not authoritative, 1
locator
      192.2.101.70 pri/wei=10/10 lpR
112248: Jul  8 18:06:37.862: [XTR] LISP-0: Remote EID IID 4100 prefix 192.2.31.3/32,
Change state to complete (sources: <map-rep>, state: reused, rlocs: 0).
```

Once the LISP resolution is complete, an update is sent to RIB and to route watch informing of the event completion.

```
// send update to RIB

112249: Jul  8 18:06:37.862: [XTR] LISP: RIB Watch Group default 192.2.101.70/32 ,
created.
112250: Jul  8 18:06:37.862: [XTR] LISP: RIB Watch Group default 192.2.101.70/32 ,
scheduling RIB update.
112255: Jul  8 18:06:37.863: [XTR] LISP: RIB Watch Group default 192.2.101.70/32 ,
installing in RIB.
112256: Jul  8 18:06:37.863: [XTR] LISP-0: Remote shrRLOC 192.2.101.70, Reachability
notification, up* allow* remote.
112257: Jul  8 18:06:37.863: [XTR] LISP-0: Remote EID IID 4100 prefix 192.2.31.3/32, No
more RLOCs pending rwatch update, schedule deferred fwd update (sources: <map-rep>,
state: complete, rlocs: 1).

// send update back to rwatch and mark prefix as LISP learnt

112258: Jul  8 18:06:37.864: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Received LISP notification, masklen = 32.
112259: Jul  8 18:06:37.864: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Prefix status set to: LISP.
```

At the end of this process, Edge2 knows exactly where the source is located and what the RLOC IP address. Finally, the RPF interface for the (S,G) entry is updated to point to the source RLOC for 192.2.31.3, which is 192.2.101.70.

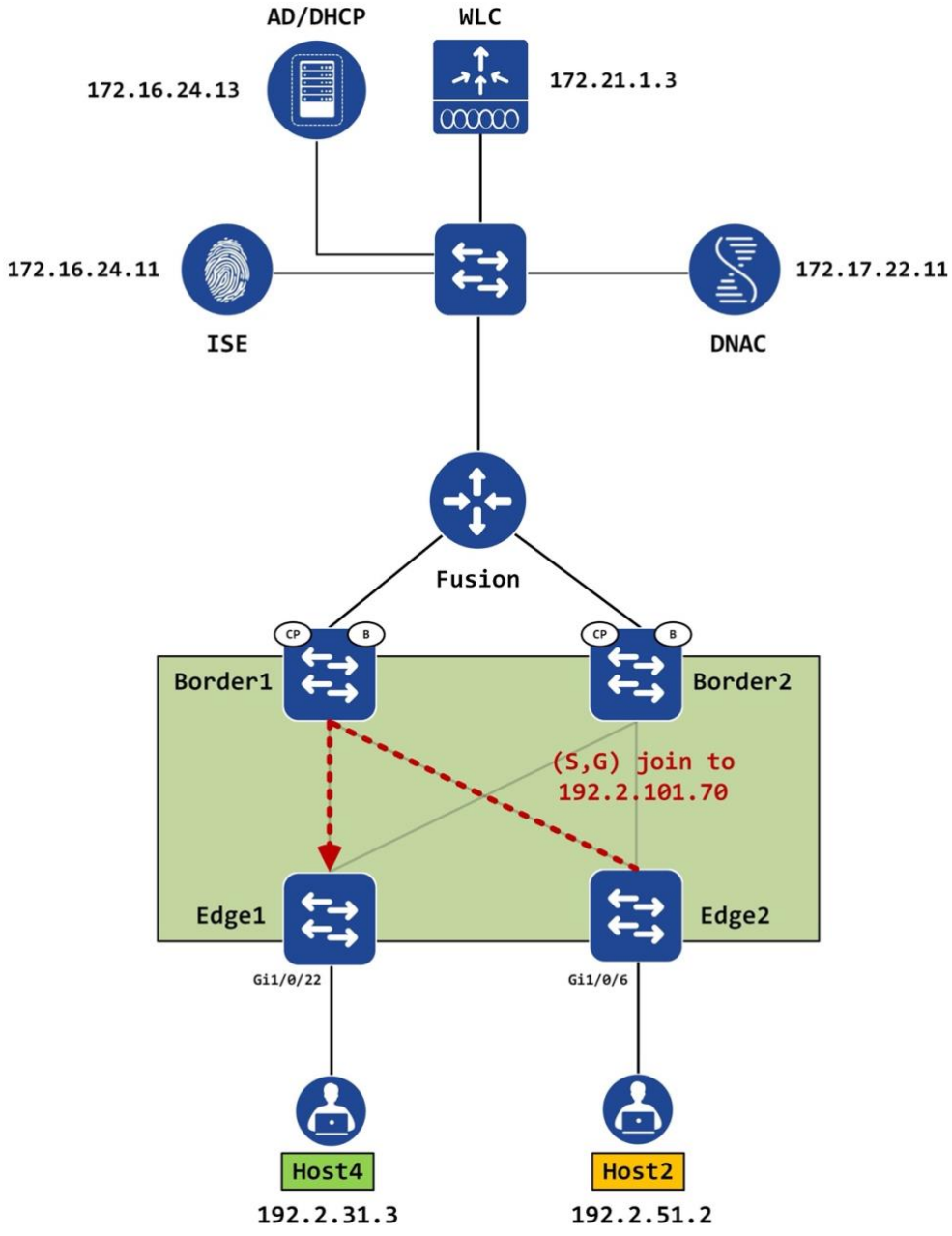
```
// update RPF on (S,G) entry to point to LISP interface/RLOC IP

112268: Jul  8 18:06:37.865: MRT(2): (192.2.31.3,239.100.100.100), RPF change from
/0.0.0.0 to LISP0.4100/192.2.101.70
112281: Jul  8 18:06:37.866: MRT(2): RPF change 192.2.101.70/LISP0.4100 for
(192.2.31.3/32, 239.100.100.100)
```


Aninda Chatterjee (aninchat@cisco.com)

At this point, a SPT cutover is triggered as well so that the last hop router is connected to the first hop router via the shortest path tree. The last hop router triggers this (Edge2, in our case), as can be seen from the below logs from Edge2:

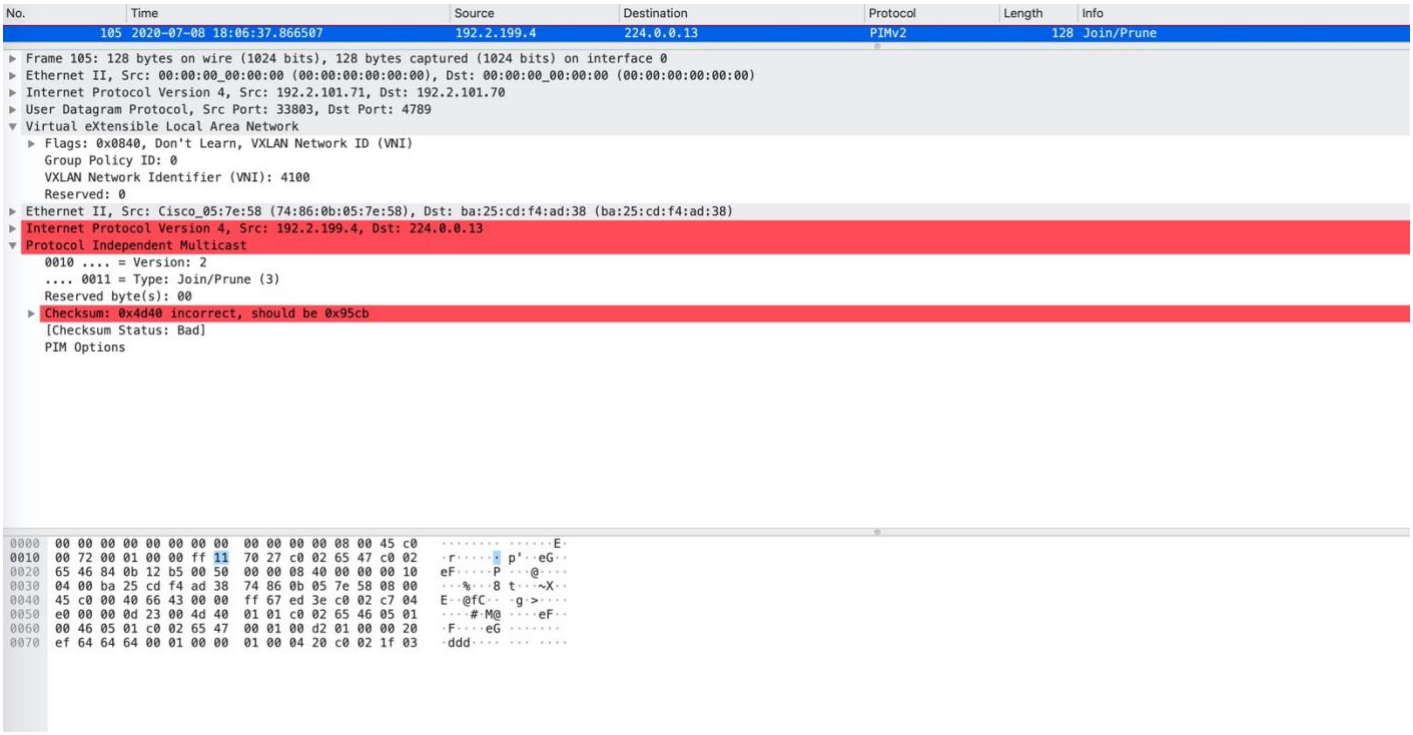
```
// trigger SPT cutover, send S-bit join to source RLOC

112278: Jul  8 18:06:37.865: MRT(2): Set the T-flag for (192.2.31.3, 239.100.100.100)
112285: Jul  8 18:06:37.866: MRT(2): Start 0x7F8195C83278 [1579] send join timer (*,
239.100.100.100) delay 0 msec , rpf_nbr 192.2.101.65/0.0.0.0
112286: Jul  8 18:06:37.866: PIM(2): Building Join/Prune packet for nbr 192.2.101.70
112287: Jul  8 18:06:37.866: PIM(2):  Adding v2 (192.2.31.3/32, 239.100.100.100), S-bit
Join
112288: Jul  8 18:06:37.866: PIM(2): Adding LISP Multicast transport attribute in
join/prune to 192.2.101.70 (LISP0.4100)
112289: Jul  8 18:06:37.866: PIM(2): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```

The (S,G) join is sent to 192.2.101.70, as seen from the above logs. This is the RLOC IP address that was resolved via route watch earlier.



Aninda Chatterjee (aninchat@cisco.com)

It is important to look at this packet because it contains a lot of good information.



Wireshark is unable to decode the capture fully and erroneously reports an invalid checksum. Again, this is because of the new join attributes that are added in the PIm options.

Let’s decode the actual hex dump to understand this better. The hex dump is:

```
0000   00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 c0
0010   00 72 00 01 00 00 ff 11 70 27 c0 02 65 47 c0 02
0020   65 46 84 0b 12 b5 00 50 00 00 08 40 00 00 00 10
0030   04 00 ba 25 cd f4 ad 38 74 86 0b 05 7e 58 08 00
0040   45 c0 00 40 66 43 00 00 ff 67 ed 3e c0 02 c7 04
0050   e0 00 00 0d 23 00 4d 40 01 01 c0 02 65 46 05 01
0060   00 46 05 01 c0 02 65 47 00 01 00 d2 01 00 00 20
0070   ef 64 64 64 00 01 00 00 01 00 04 20 c0 02 1f 03
```

23 – PIM version of 2 and type of 3 (implying a join/prune).
c0 02 65 46 – 192.2.101.70 (the destination this join is sent to).
05 01 00 – this is the transport attribute. The first 8 bits give you the fbit, the ebit and the type. The first bit is the fbit, the second bit is the ebit and the last 6 bits form the type of attribute (should be set to 5 to indicate it is the transport attribute). So in this case, we have 0000 0101 which gives us a fbit of 0, an ebit of 0 and a type of 5. The next 8 bits specify the length which is 01 in this case. The last 8 bits specify the transport – this is very important because this dictates whether the underlay is using multicast or headend replication (unicast). This is set to 0 for multicast and 1 for unicast. In our case, as you can see, it is 00.
46 05 01 - this is the receiver RLOC attribute. This is primarily used to provide the RLOC IP address of the ETR. The first 8 bits are again used for fbit, ebit and the type. In this case, we have a hex of 46 which translates to 0100 0110. The first bit is the fbit which is 0, the second bit is the ebit which is 1 (implying an end of attributes and that no more attributes follow), the last 6 bits form the type which is 000110, giving us a value of 6, which is the receiver RLOC attribute. The next 8 bits are used for length (the value is in octets and is used to specify the length of the actual attribute value + 1 to account for address family). It is 5 because the RLOC

value is an IP address (4 octets) and 1 octet will be used for the address family. Thus, 01 is the address family which is IPv4 (address family numbers can be found on the official IANA page and PIM simply follows that till 127 - https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml). **c0 02 65 47** – this is the RLOC address. Decoding this, you get 192.2.101.71, which is the loopback0 IP address of Edge2.

Thus, with this packet, Edge2 is able to inform Edge1 that the transport it should use is multicast and what the RLOC address of the receiver ETR is.

This (S,G) join is processed on Edge1:

```
// Edge1 processing S-bit join from Edge2

452582: Jul  8 18:06:37.867: PIM(2): J/P Transport Attribute, Transport Type: Multicast
452583: Jul  8 18:06:37.867: PIM(2): Join-list: (192.2.31.3/32, 239.100.100.100), S-bit
set
452584: Jul  8 18:06:37.867: [XTR] LISP-0: Dyn EID mcast join IID 4100 192.2.31.3
192.2.101.71 Created.
452585: Jul  8 18:06:37.867: PIM(2): Update LISP0.4100/192.2.101.71 to (192.2.31.3,
239.100.100.100), Forward state, by PIM SG Join
```

Finally, the RP is pruned from the path in the overlay because there is no requirement for it to forward traffic in the overlay anymore (post SPT cutover). The underlay SSM is what will be used to forward the traffic through the fabric.

```
// Edge2 pruning the RP in overlay

112330: Jul  8 18:06:37.875: PIM(2): Building Join/Prune packet for nbr 192.2.101.65
112331: Jul  8 18:06:37.875: PIM(2):  Adding v2 (192.2.199.1/32, 239.100.100.100), WC-
bit, RPT-bit, S-bit Join
112332: Jul  8 18:06:37.875: PIM(2):  Adding v2 (192.2.31.3/32, 239.100.100.100), RPT-
bit, S-bit Prune
112333: Jul  8 18:06:37.875: PIM(2): Adding LISP Multicast transport attribute in
join/prune to 192.2.101.65 (LISP0.4100)
112334: Jul  8 18:06:37.875: PIM(2): Send v2 join/prune to 192.2.101.65 (LISP0.4100)
```


Aninda Chatterjee (aninchat@cisco.com)

Understanding the final MRIB state on each device

Edge1:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 00:01:30/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65, LISP: [192.2.101.65, 232.0.3.1]
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 00:01:30/00:01:30, flags: FTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.1), Forward/Sparse, 00:01:30/stopped, p
      192.2.101.71, 00:01:28/00:02:59
```

```
Edge1#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 00:01:41/00:02:47, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:01:41/00:02:47, p

(192.2.101.65, 232.0.3.1), 00:01:41/00:01:18, flags: sT
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 192.2.101.68
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 00:01:41/stopped
```

In the overlay, Edge1 has both the (*,G) and the (S,G) for group 239.100.100.100. The source tree is used to forward the traffic natively.

Since we are using native multicast here, the multicast traffic will get encapsulated with multicast itself – this is where the underlay SSM comes in. In the underlay, we have a (S,G) where Edge1 RLOC address is the source. The incoming interface is null, we Edge1 is doing the encapsulation. The outgoing interface is the interface that connects to Border1.

Here’s a packet capture of the packet leaving Edge1 and going towards Border1:

Aninda Chatterjee (aninchat@cisco.com)

No.	Time	Source	Destination	Protocol	Length	Info
131	2020-07-08 18:06:38.743157	192.2.31.3	239.100.100.100	UDP	124	8910 → 8910 Len=32
135	2020-07-08 18:06:39.258045	192.2.31.3	239.100.100.100	UDP	124	8910 → 8910 Len=32
137	2020-07-08 18:06:39.772874	192.2.31.3	239.100.100.100	UDP	124	8910 → 8910 Len=32
139	2020-07-08 18:06:40.287681	192.2.31.3	239.100.100.100	UDP	124	8910 → 8910 Len=32
141	2020-07-08 18:06:40.802531	192.2.31.3	239.100.100.100	UDP	124	8910 → 8910 Len=32

```
▶ Frame 137: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface 0
▶ Ethernet II, Src: Cisco_9f:1f:d6 (00:00:0c:9f:1f:d6), Dst: IPv4mcast_03:01 (01:00:5e:00:03:01)
▼ Internet Protocol Version 4, Src: 192.2.101.70, Dst: 232.0.3.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 110
    Identification: 0x1f3e (7998)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 32
    Protocol: UDP (17)
    Header checksum: 0x2af7 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.2.101.70
    Destination: 232.0.3.1
▶ User Datagram Protocol, Src Port: 65348, Dst Port: 4789
▼ Virtual eXtensible Local Area Network
  ▶ Flags: 0x8800, GBP Extension, VXLAN Network ID (VNI)
    Group Policy ID: 20
    VXLAN Network Identifier (VNI): 4100
    Reserved: 0
▶ Ethernet II, Src: Cisco_9f:00:00 (00:00:0c:9f:00:00), Dst: IPv4mcast_64:64:64 (01:00:5e:64:64:64)
▼ Internet Protocol Version 4, Src: 192.2.31.3, Dst: 239.100.100.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x6515 (25877)
  ▶ Flags: 0x0000
    Time to live: 32
    Protocol: UDP (17)
    Header checksum: 0x02ce [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.2.31.3
    Destination: 239.100.100.100
▶ User Datagram Protocol, Src Port: 8910, Dst Port: 8910
▶ Data (32 bytes)
```


Aninda Chatterjee (aninchat@cisco.com)

Border1:

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1w2d/00:02:38, RP 192.2.199.1, flags: Sp
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.65, 232.0.3.1), Forward/Sparse, 00:18:24/stopped, p
      192.2.101.71, 00:18:24/00:02:38

(192.2.31.3, 239.100.100.100), 00:01:51/00:01:09, flags: PT
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70, 232.0.3.1]
  Outgoing interface list: Null
```

```
Border1#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 00:02:07/00:03:22, flags: sTp
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 192.2.101.69
  lisp eid refcnt: 1
  Outgoing interface list:
    GigabitEthernet1/0/3, Forward/Sparse, 00:02:05/00:03:22, p
    Null0, Forward/Dense, 00:02:07/stopped

(192.2.101.65, 232.0.3.1), 00:18:40/00:03:27, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:02:07/00:03:19, p
    GigabitEthernet1/0/3, Forward/Sparse, 00:18:40/00:03:27, p
```

In the overlay, the (S,G) is pruned on Border1. This is because there is no requirement to forward in the overlay. The packet will come encapsulated on the border and it will purely forward based on the underlay SSM.

In the underlay, we have a (S,G) for Edge1 as the source and the derived underlay group. The outgoing interface list contains gig1/0/3 which points to Edge2.

Aninda Chatterjee (aninchat@cisco.com)

Edge2:

```
Edge2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 1w2d/stopped, RP 192.2.199.1, flags: SJC
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65, LISP: [192.2.101.65, 232.0.3.1]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 1w2d/00:02:19

(192.2.31.3, 239.100.100.100), 00:02:17/00:00:42, flags: JT
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70, 232.0.3.1]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 00:02:17/00:02:19
```

```
Edge2#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 00:02:31/00:00:28, flags: sT
  Incoming interface: GigabitEthernet1/0/3, RPF nbr 192.2.101.72
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 00:02:31/stopped

(192.2.101.65, 232.0.3.1), 00:19:05/00:01:54, flags: sT
  Incoming interface: GigabitEthernet1/0/3, RPF nbr 192.2.101.72
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 00:19:05/stopped
```

On Edge2, remember, the packet would be coming encapsulated. It will first hit the underlay SSM entry – there is an entry for (192.2.101.70, 232.0.3.1) with an outgoing interface list of Null0. This is installed as a decapsulation action in hardware and the outer header is decapsulated.

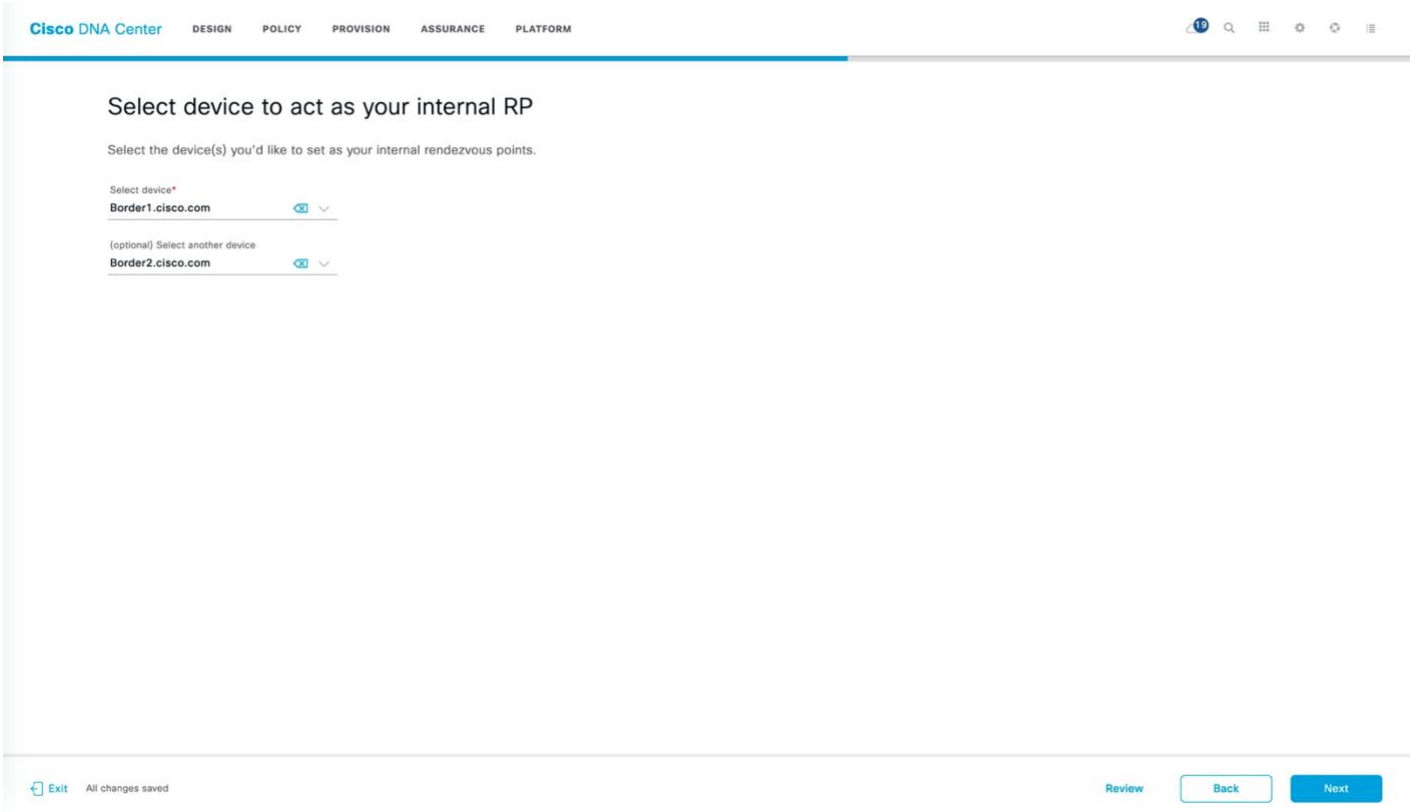
Post decapsulation, the packet would be recirculated again and we do another lookup for the inner multicast packet, now hitting the VRF specific MRIB, which contains an entry for (192.2.31.3, 239.100.100.100). The packet will now be sent to the receiver.

Aninda Chatterjee (aninchat@cisco.com)

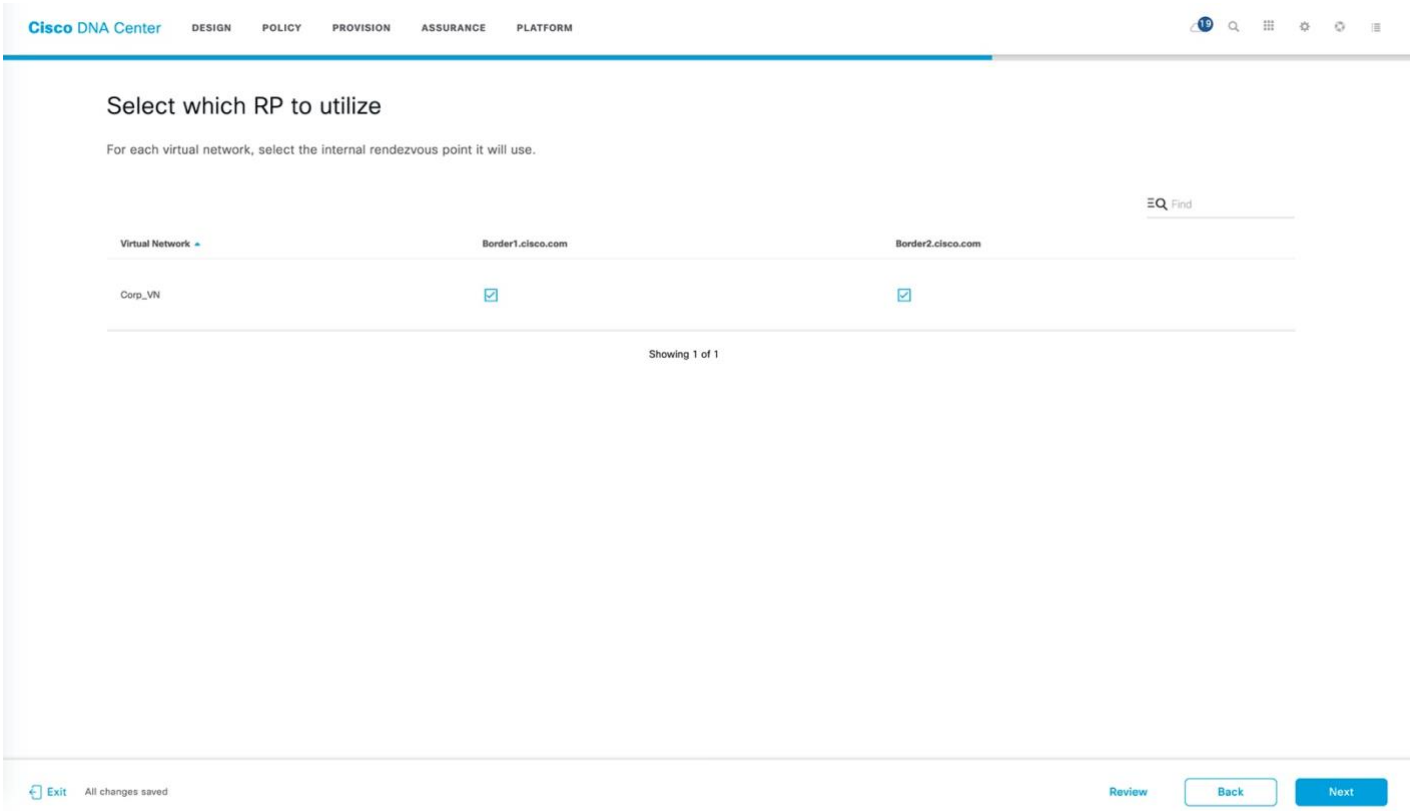
Native multicast – ASM with dual RPs

Configuring dual RPs via DNAC

The configuration is largely the same from DNAC – the only difference is that you need to choose a second RP as well, which will be Border2 in our case.

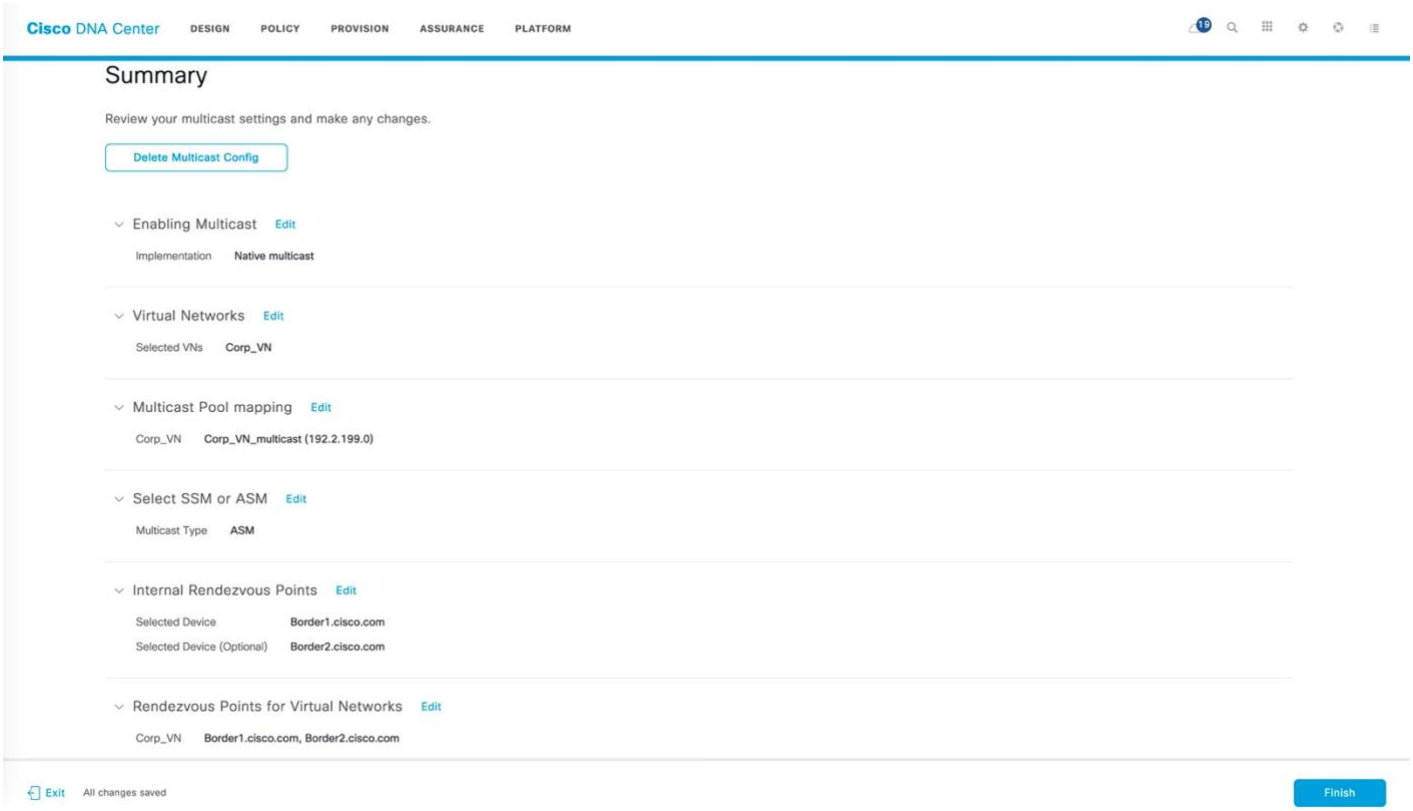


You then choose these RPs for the VN you want to enable for multicast:



Aninda Chatterjee (aninchat@cisco.com)

Finally, a summary page will show you your choice like before:



The following configuration gets pushed to the fabric devices as part of this workflow:

Edge1:

```
// multicast routing enabled for Corp_VN VRF

004860: Jul 24 13:28:19.279: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast-routing vrf Corp_VN

// transport configured to be multicast and underlay SSM group range is defined

004861: Jul 24 13:28:19.297: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface LISP0.4100
004862: Jul 24 13:28:19.327: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp transport multicast
004863: Jul 24 13:28:19.350: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp core-group-range 232.0.0.1 1000

// hsot facing SVIs enabled for IGMP

004864: Jul 24 13:28:19.365: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan1029
004865: Jul 24 13:28:19.382: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp explicit-tracking
004866: Jul 24 13:28:19.394: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim passive
004867: Jul 24 13:28:19.418: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp version 3
004868: Jul 24 13:28:19.424: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:exit
```

```
004869: Jul 24 13:28:19.437: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan1030
004870: Jul 24 13:28:19.457: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp explicit-tracking
004871: Jul 24 13:28:19.469: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim passive
004872: Jul 24 13:28:19.492: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp version 3
004873: Jul 24 13:28:19.499: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:exit

// loopback4100 created for multicast

004879: Jul 24 13:28:19.588: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4100
004880: Jul 24 13:28:19.600: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
004881: Jul 24 13:28:19.618: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.3 255.255.255.255
004882: Jul 24 13:28:19.630: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// RP configured and loopback4100 used as source
// SSM default range will be used

004883: Jul 24 13:28:19.659: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN rp-address 192.2.199.1
004884: Jul 24 13:28:19.708: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN register-source Loopback4100
004885: Jul 24 13:28:19.716: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN ssm default

// loopback4100s IP mapped under LISP database

004886: Jul 24 13:28:19.727: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:router lisp
004913: Jul 24 13:28:20.731: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:instance-id 4100
004914: Jul 24 13:28:20.735: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:service ipv4
004919: Jul 24 13:28:20.886: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:database-mapping 192.2.199.3/32  locator-set rloc_70ac0729-86f6-44d0-a1c9-
254b98507c24

// multipath enabled for multicast

005323: Jul 24 13:28:28.542: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast vrf Corp_VN multipath
```

Similar configuration is pushed on Edge2 as well, so we're not going to cover that. Let's look at Border1s configuration now.

Aninda Chatterjee (aninchat@cisco.com)

Border1:

```
// multicast routing enabled for Corp_VN VRF

Jul 24 2020 13:27:44.222 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast-routing vrf Corp_VN

// transport configured to be multicast and underlay SSM group range is defined

Jul 24 2020 13:27:44.253 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface LISP0.4100
Jul 24 2020 13:27:44.311 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp transport multicast
Jul 24 2020 13:27:44.367 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp core-group-range 232.0.0.1 1000

// L3 handoff SVI in the same VRF is enabled with PIM SM

Jul 24 2020 13:27:44.397 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan3014
Jul 24 2020 13:27:44.420 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// loopback4100 configured in same VRF and enabled for PIM SM

Jul 24 2020 13:27:44.496 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4100
Jul 24 2020 13:27:44.543 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
Jul 24 2020 13:27:44.613 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.1 255.255.255.255
Jul 24 2020 13:27:44.645 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// loopback 4600 enabled for PIM-SM. This will be used to form MSDP peering between the
two RPs

Jul 24 2020 13:27:44.688 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4600
Jul 24 2020 13:27:44.728 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
Jul 24 2020 13:27:44.791 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.7 255.255.255.255
Jul 24 2020 13:27:44.817 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// RP defined to be loopback4100s IP

Jul 24 2020 13:27:44.881 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN rp-address 192.2.199.1
Jul 24 2020 13:27:45.013 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN register-source Loopback4100
Jul 24 2020 13:27:45.057 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN ssm default

//  MSDP peering configured. Border1 and Border2 will be the peers
```

```
Jul 24 2020 13:27:45.100 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip msdp vrf Corp_VN cache-sa-state
Jul 24 2020 13:27:45.156 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip msdp vrf Corp_VN originator-id Loopback4600
Jul 24 2020 13:27:45.233 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip msdp vrf Corp_VN peer 192.2.199.6 connect-source Loopback4600
Jul 24 2020 13:27:45.284 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip msdp originator-id Loopback4600

// the multicast pool used for loopback creation is advertised into BGP

Jul 24 2020 13:27:45.313 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:router bgp 65003
Jul 24 2020 13:27:45.357 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:address-family ipv4 vrf Corp_VN
Jul 24 2020 13:27:45.421 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:aggregate-address 192.2.199.0 255.255.255.0 summary-only
Jul 24 2020 13:27:45.483 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:network 192.2.199.1 mask 255.255.255.255

// both loopbacks mapped to LISP database

Jul 24 2020 13:27:46.384 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:instance-id 4100
Jul 24 2020 13:27:46.688 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:database-mapping 192.2.199.7/32  locator-set rloc_23bd3133-ba7b-4058-9552-
e37e69cbc843
Jul 24 2020 13:27:46.811 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:database-mapping 192.2.199.1/32  locator-set rloc_23bd3133-ba7b-4058-9552-
e37e69cbc843

// since Border1 is a control-plane as well, allow multicast pool subnet under LISP
site

Jul 24 2020 13:27:55.255 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:site site_uci
Jul 24 2020 13:27:55.329 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:authentication-key uci
Jul 24 2020 13:27:55.462 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:eid-record instance-id 4100 192.2.199.0/24  accept-more-specifics
```

Border2 will have similar configurations pushed to it so we're not going to look at that in detail.

At the end of this, Border1 and Border2 must be MSDP peers. Let's confirm that:

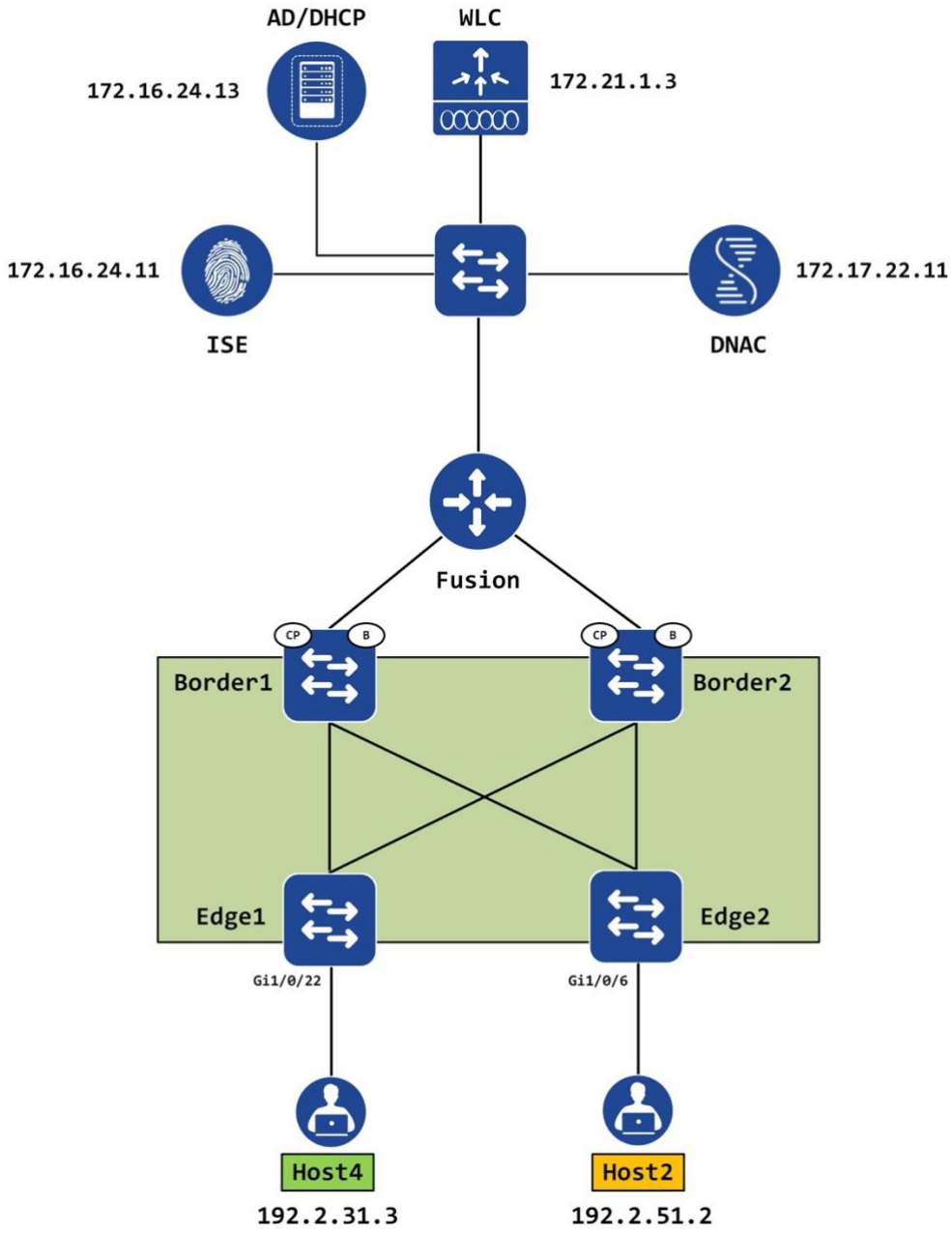
```
Border1#show ip msdp vrf Corp_VN peer
MSDP Peer 192.2.199.6 (?), AS 65003
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback4600 (192.2.199.7)
    Uptime(Downtime): 01:46:14, Messages sent/received: 107/107
    Output messages discarded: 0
    Connection and counters cleared 01:47:26 ago

*snip*
```


Aninda Chatterjee (aninchat@cisco.com)

```
Border2#show ip msdp vrf Corp_VN peer
MSDP Peer 192.2.199.7 (?), AS 65003
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback4600 (192.2.199.6)
    Uptime(Downtime): 01:44:26, Messages sent/received: 105/105
    Output messages discarded: 0
    Connection and counters cleared 01:45:43 ago

*snip*
```

Receiver join and building the shared tree

We'll use the following topology for this (Host2 is back!):



Both Border1 and Border2 are RPs in this case, as seen from the configurations pushed earlier. Loopback4100 is essentially a floating RP address between the two boxes, constituting an anycast RP design. The two borders are also MSDP peers as seen earlier.

When Host2 sends an IGMP report to join the group 239.50.50.50, this will be sent as a (*,G) join to the RP address 192.2.199.1. Now, both the borders own this and Edge2 should have an equal cost paths to this destination:

```
Edge2#show ip cef vrf Corp_VN 192.2.199.1
192.2.199.1/32
  nexthop 192.2.101.65 LISP0.4100
  nexthop 192.2.101.66 LISP0.4100
```


Aninda Chatterjee (aninchat@cisco.com)

Naturally, only one will be chosen out of this. In our case, the join was sent to Border1. This is why Border1 has the following MRIB table, while Border2 has no entry for this group:

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:00:16/00:03:13, RP 192.2.199.1, flags: Sp
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.65, 232.0.3.1), Forward/Sparse, 00:00:16/stopped, p
      192.2.101.71, 00:00:16/00:03:13
```

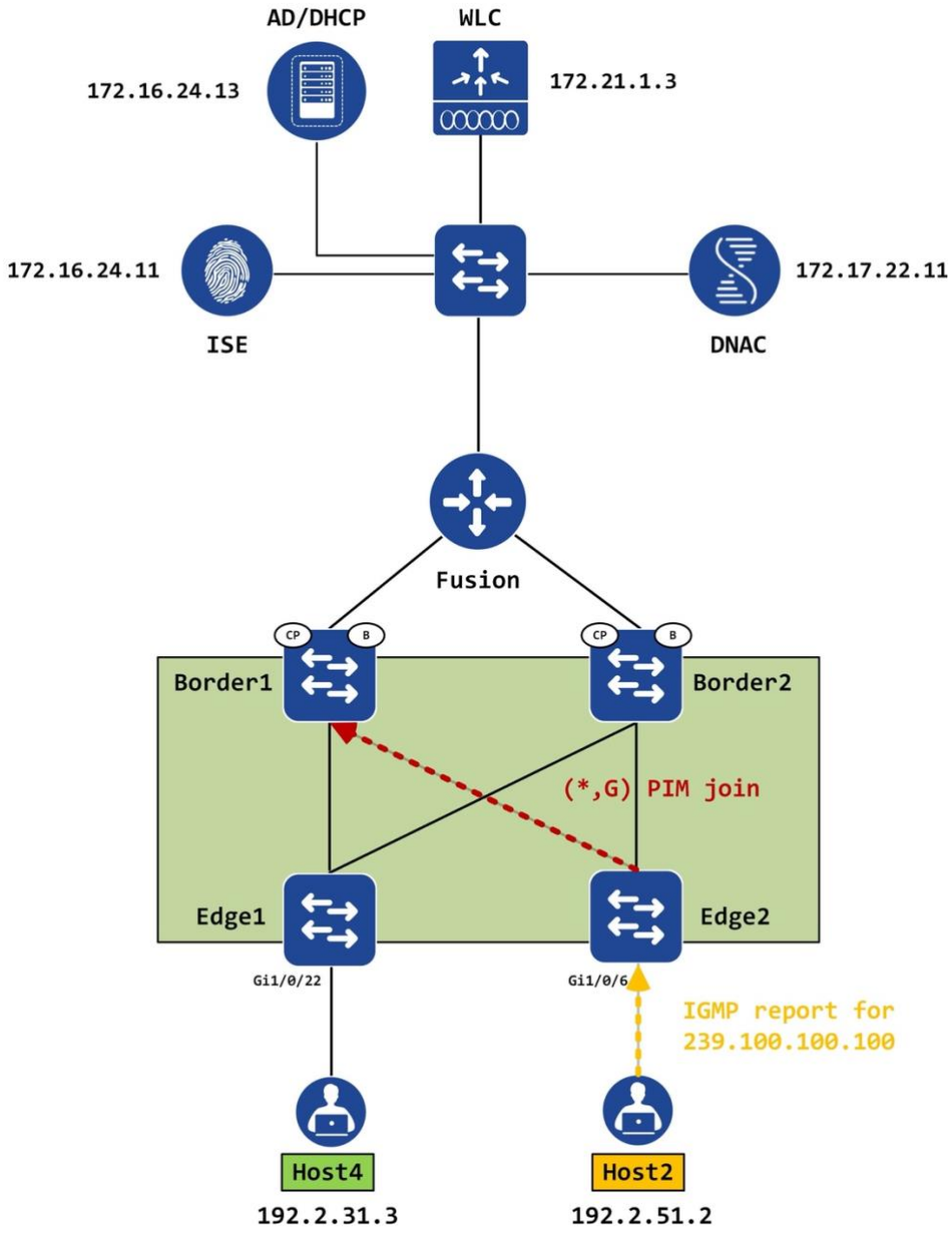
```
Border2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
Group 239.100.100.100 not found
```

A corresponding entry in the underlay is created as well:

```
Border1#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.65, 232.0.3.1), 00:02:46/00:02:40, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/3, Forward/Sparse, 00:02:46/00:02:40, p
```

```
Border2#show ip mroute 232.0.3.1 verbose
Group 232.0.3.1 not found
```


Aninda Chatterjee (aninchat@cisco.com)



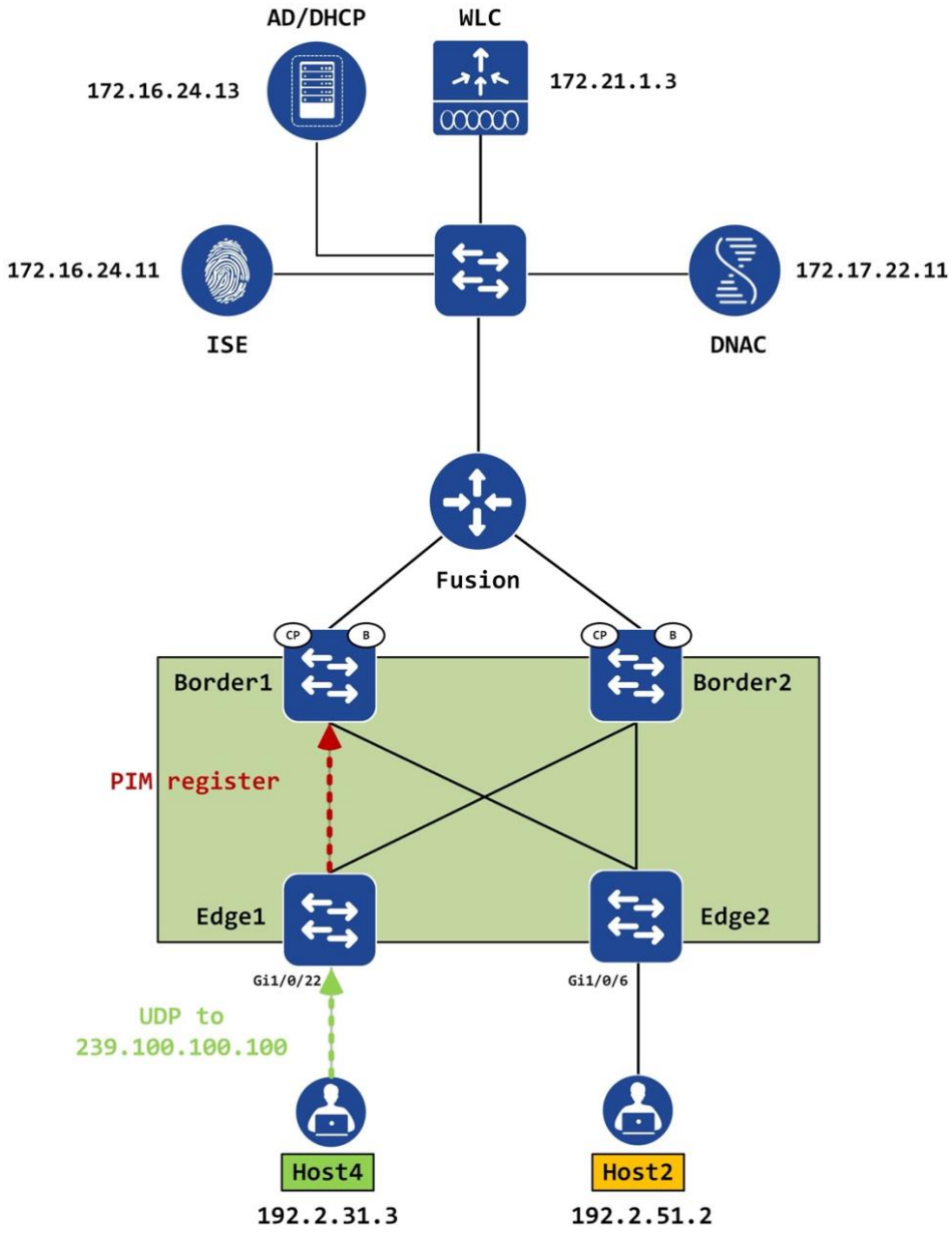
Why does Border2 not maintain an entry for this? Remember, MSDP does not sync (*,G) routes – the whole purpose of MSDP is to sync active sources and thus, only the (S,G) entries.

Source start and building the SPT

Once the source starts, the process is largely the same as what we've seen earlier. Edge1 will register the source to the RP. Now, just like Edge2, Edge1 also has equal cost paths to the RPs IP address:

```
Edge1#show ip cef vrf Corp_VN 192.2.199.1
192.2.199.1/32
  nexthop 192.2.101.65 LISP0.4100
  nexthop 192.2.101.66 LISP0.4100
```

It will either send the register to Border1 or Border2. In our case, it sends it to Border1.

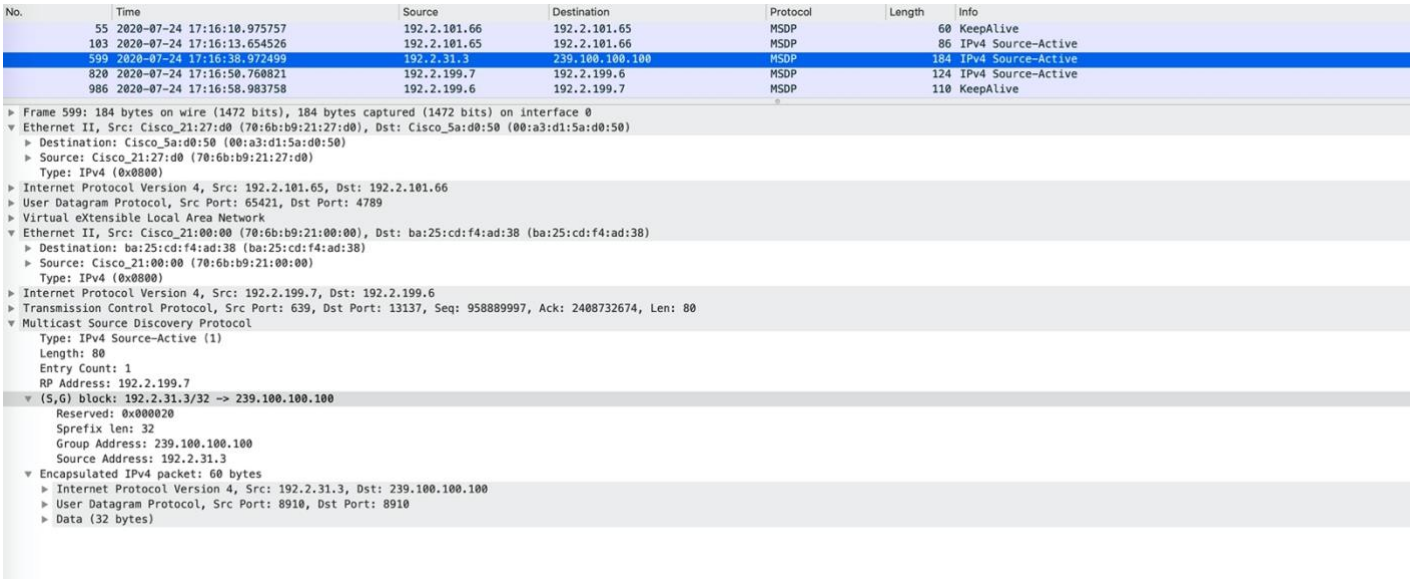


Border1 and Border2 are MSDP peers and thus, once Border1 learns of the source, it must inform Border2 as well.

Debugs from Border1:

```
// register received from Edge1

Jul 24 17:16:38.957: PIM(117): Received v2 Register on LISP0.4100 from 192.2.199.3
Jul 24 17:16:38.957:      for 192.2.31.3, group 239.100.100.100

// route watch triggered for source 192.2.31.3

Jul 24 17:16:38.958: MCAST-RWATCH: -af IPv4 -topo 8075: Route watch topology found.
Jul 24 17:16:38.958: MCAST-RWATCH: -af IPv4 -topo 8075 -prefix 192.2.31.3/32: Route
watch prefix not found.
Jul 24 17:16:38.958: MCAST-RWATCH: -af IPv4 -topo 8075 -prefix 192.2.31.3/32: Route
watch prefix created.
Jul 24 17:16:38.959: MCAST-RWATCH: -af IPv4 -topo 8075 -prefix 192.2.31.3/32: Started
watching prefix in RIB.
Jul 24 17:16:38.959: MCAST-RWATCH: -af IPv4 -topo 8075 -prefix 192.2.31.3/32: Started
watching prefix in LISP.

// (S,G) entry created

Jul 24 17:16:38.964: MRT(117): Create (192.2.31.3,239.100.100.100), RPF (unknown,
0.0.0.0, 0/0)
Jul 24 17:16:38.967: MRT(117): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 232.0.3.1 Outer-source: 192.2.101.65 Successful
Jul 24 17:16:38.970: MRT(117): Add LISP0.4100/232.0.3.1/192.2.101.65 to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built

// send MSDP SA encapsulated message to peer

Jul 24 17:16:38.971: MSDP(117): 192.2.199.6: Send 60-byte SA encapsulated data for
(192.2.31.3, 239.100.100.100), RP 192.2.199.7
```

This packet can also be seen in the capture below:



This packet is very interesting and actually has multiple layers of encapsulation. Just like a register, the actual multicast data can also be carried behind a MSDP packet

which is what we see here (inner most IP packet). Above that is the actual MSDP header which includes the RP address (this is the address used for MSDP peering) and the (S,G) that it is advertising to its peer.

Remember that the reachability is over VXLAN which is why a final VXLAN header is added as well.

Border2 should now have this (S,G) entry in its SA cache:

```
Border2#show ip msdp vrf Corp_VN sa-cache
MSDP Source-Active Cache - 1 entries
(192.2.31.3, 239.100.100.100), RP 192.2.199.7, BGP/AS 65003, 21:34:00/00:05:15, Peer
192.2.199.7
```

However, has Border2 installed this in its multicast routing table?

```
Border2#show ip mroute vrf Corp_VN 239.100.100.100
Group 239.100.100.100 not found
```

Still nothing! Why? Well, because MSDP rules are very simple – if both the (*,G) and the (S,G) are registered on Border1, there is no need for Border2 to take part in the process.

The source tree is built in the exact same way as we've seen earlier and eventually SPT cutover will be triggered as well. No changes in how all of that happens so we're not going to cover that again.

Aninda Chatterjee (aninchat@cisco.com)

Understanding the MSDP process

MSDP really kicks in when a peer gets a SA message and it has a (*,G) entry in its multicast routing table. The RFC dictates that this peer must now trigger a (S,G) join towards the source. From the RFC (RFC 3618) directly:

"When an MSDP peer which is also an RP for its own domain receives a new SA message, it determines if there are any group members within the domain interested in any group described by an (Source, Group), or (S,G) entry within the SA message. That is, the RP checks for a (,G) entry with a non-empty outgoing interface list; this implies that some system in the domain is interested in the group. In this case, the RP triggers a (S,G) join event towards the data source as if a Join/Prune message was received addressed to the RP itself."*

So, let's try and force this in our setup. On Edge2, I've statically set the RPs floating IP address (192.2.199.1) to be mapped to Border2 only.

```
Edge2(config)#router lisp
Edge2(config-router-lisp)#instance-id 4100
Edge2(config-lisp-inst)#service ipv4
Edge2(config-lisp-inst-srv-ipv4)# map-cache 192.2.199.1/32 192.2.101.66 priority 5
weight 10
```

On Edge2, we no longer have equal cost paths to this destination anymore and now when Edge2 has to send anything to 192.2.199.1, it will send it to Border2. Let's start our receiver again. This time, the (*,G) is sent to Border2:

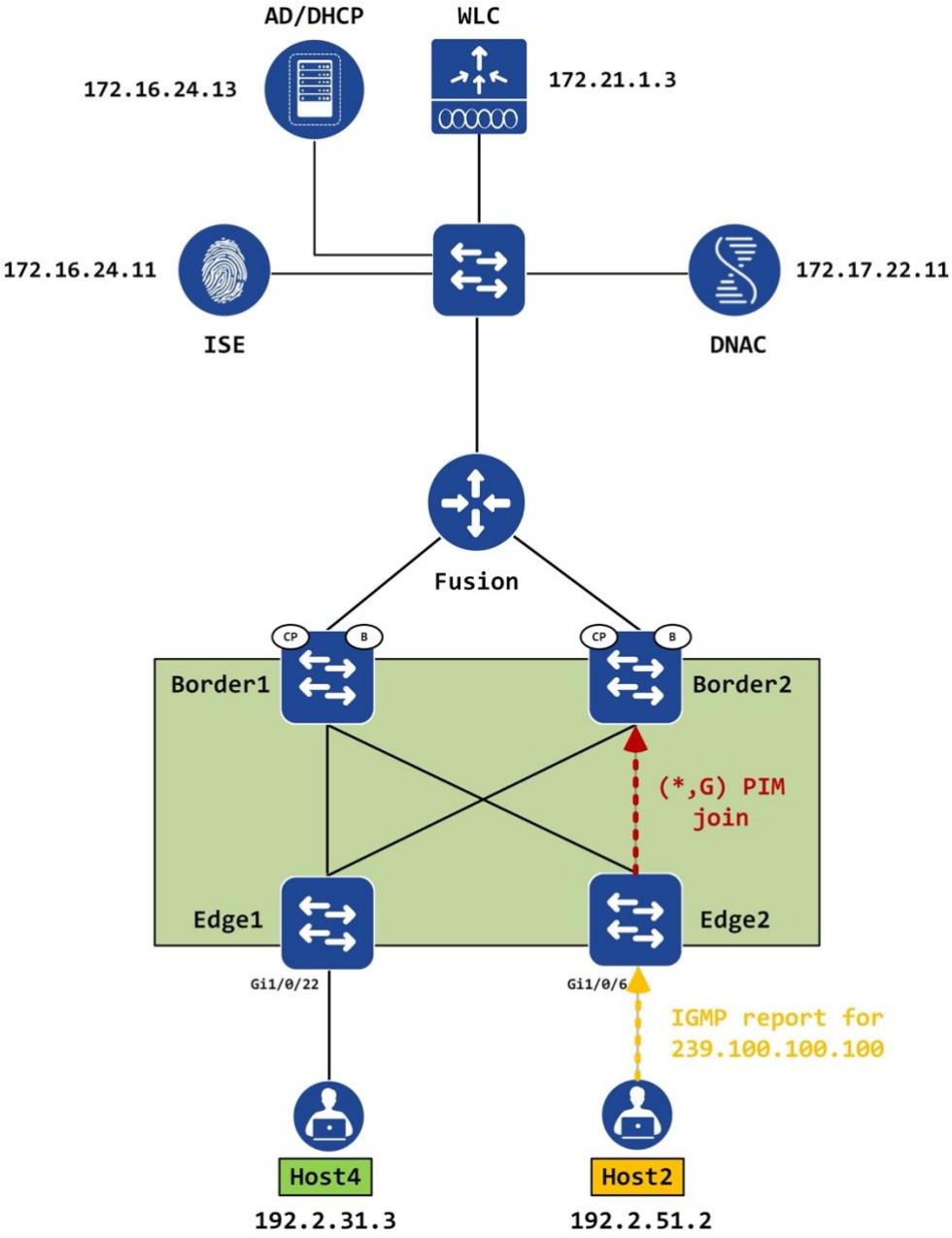
```
Edge2#show ip cef vrf Corp_VN 192.2.199.1
192.2.199.1/32
  nexthop 192.2.101.66 LISP0.4100
```

```
Border2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.100.100.100), 00:00:12/00:03:17, RP 192.2.199.1, flags: Sp
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.66, 232.0.3.1), Forward/Sparse, 00:00:12/stopped, p
      192.2.101.71, 00:00:12/00:03:17
```


Aninda Chatterjee (aninchat@cisco.com)

As expected, Border1 has no idea about this group yet:

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
Group 239.100.100.100 not found
```



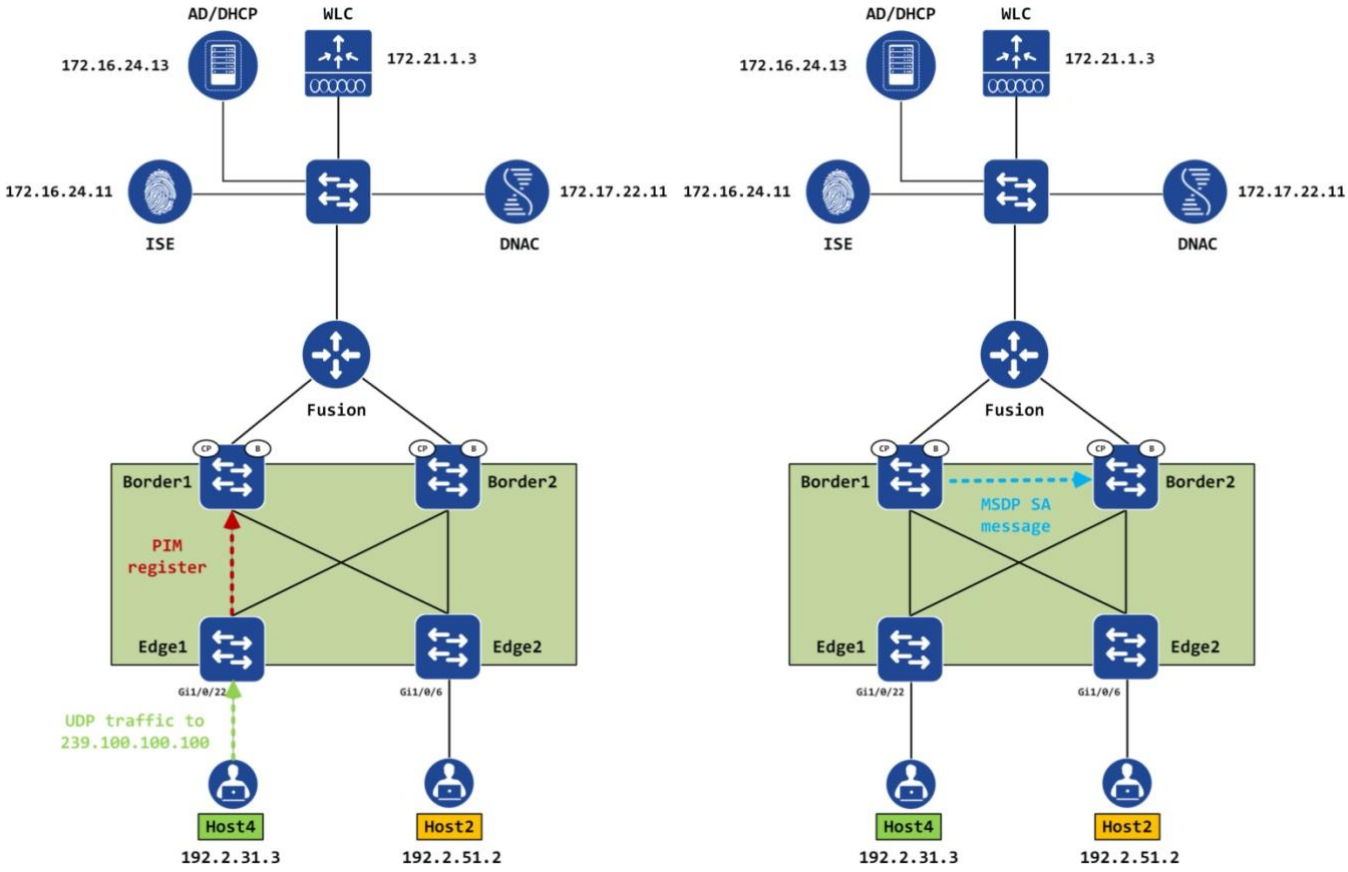
The source starts and it sends a register to Border1:

```
Jul 26 11:42:48.954: PIM(117): Received v2 Register on LISP0.4100 from 192.2.199.3
Jul 26 11:42:48.954:      for 192.2.31.3, group 239.100.100.100
```

Border1 will build a (S,G) state for this and then inform its MSDP peer (Border2) of this source learn via a SA message (just like we saw before):

Aninda Chatterjee (aninchat@cisco.com)

```
Jul 26 11:42:48.968: MSDP(117): 192.2.199.6: Send 60-byte SA encapsulated data for
(192.2.31.3, 239.100.100.100), RP 192.2.199.7
```



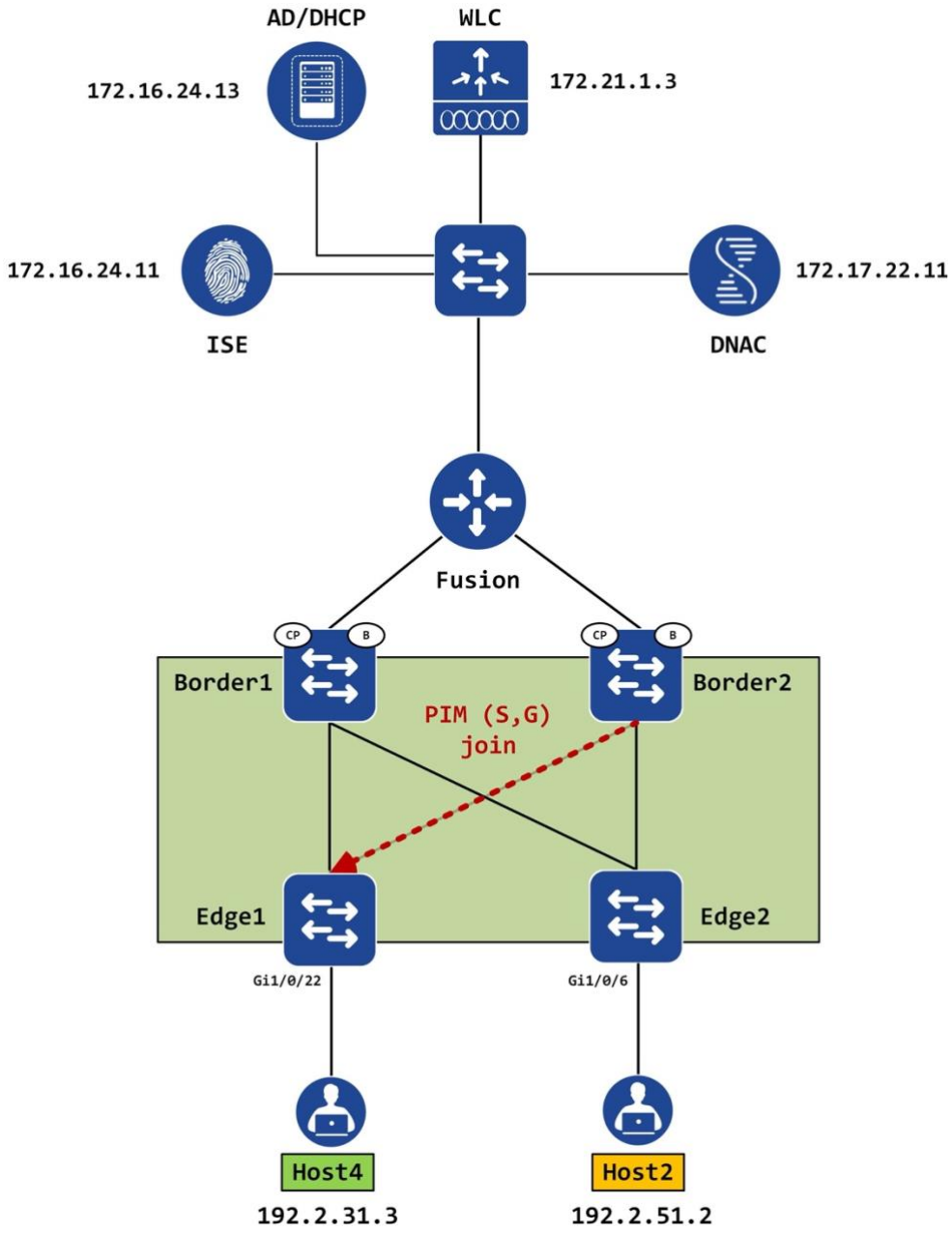
The packet format is the same as before:

No.	Time	Source	Destination	Protocol	Length	Info
263	2020-07-26 11:42:48.971040	192.2.31.3	239.100.100.100	MSDP	184	IPv4 Source-Active
517	2020-07-26 11:43:02.218749	192.2.101.65	192.2.101.66	MSDP	74	IPv4 Source-Active
724	2020-07-26 11:43:12.497565	192.2.199.6	192.2.199.7	MSDP	110	KeepAlive
800	2020-07-26 11:43:16.500111	192.2.101.66	192.2.101.65	MSDP	74	IPv4 Source-Active

```
▶ Frame 263: 184 bytes on wire (1472 bits), 184 bytes captured (1472 bits) on interface 0
▼ Ethernet II, Src: Cisco_21:27:d0 (70:6b:b9:21:27:d0), Dst: Cisco_5a:d0:50 (00:a3:d1:5a:d0:50)
  ▶ Destination: Cisco_5a:d0:50 (00:a3:d1:5a:d0:50)
  ▶ Source: Cisco_21:27:d0 (70:6b:b9:21:27:d0)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.2.101.65, Dst: 192.2.101.66
▶ User Datagram Protocol, Src Port: 65421, Dst Port: 4789
▶ Virtual eXtensible Local Area Network
▼ Ethernet II, Src: Cisco_21:00:00 (70:6b:b9:21:00:00), Dst: ba:25:cd:f4:ad:38 (ba:25:cd:f4:ad:38)
  ▶ Destination: ba:25:cd:f4:ad:38 (ba:25:cd:f4:ad:38)
  ▶ Source: Cisco_21:00:00 (70:6b:b9:21:00:00)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.2.199.7, Dst: 192.2.199.6
▶ Transmission Control Protocol, Src Port: 639, Dst Port: 13137, Seq: 958927037, Ack: 2408740312, Len: 80
▼ Multicast Source Discovery Protocol
    Type: IPv4 Source-Active (1)
    Length: 80
    Entry Count: 1
    RP Address: 192.2.199.7
  ▼ (S,G) block: 192.2.31.3/32 -> 239.100.100.100
      Reserved: 0x000020
      Sprefix len: 32
      Group Address: 239.100.100.100
      Source Address: 192.2.31.3
  ▼ Encapsulated IPv4 packet: 60 bytes
    ▶ Internet Protocol Version 4, Src: 192.2.31.3, Dst: 239.100.100.100
    ▶ User Datagram Protocol, Src Port: 8910, Dst Port: 8910
    ▶ Data (32 bytes)
```

Border2 receives this MSDP SA message. Since it has a (*,G) indicating that a receiver is interested in this group, Border2 will initiate a source join, the same way if it had received the register directly.

Aninda Chatterjee (aninchat@cisco.com)




```
// Border2 received MSDP SA message

Jul 26 11:42:48.971: MSDP(188): Received 80-byte TCP segment from 192.2.199.7
Jul 26 11:42:48.971: MSDP(188): Append 80 bytes to 0-byte msg 2890 from 192.2.199.7, qs
1
Jul 26 11:42:48.971: MSDP(188): 192.2.199.7: Received 80-byte msg 2890 from peer
Jul 26 11:42:48.971: MSDP(188): 192.2.199.7: SA TLV, len: 80, ec: 1, RP: 192.2.199.7,
with data
Jul 26 11:42:48.971: MSDP(188): 192.2.199.7: Peer RPF check passed for single peer
Jul 26 11:42:48.971: MSDP(188): (192.2.31.3/32, 239.100.100.100), accepted
Jul 26 11:42:48.971: MSDP(188): WAVL Insert SA Source 192.2.31.3 Group 239.100.100.100
RP 192.2.199.7 Successful

// route watch started for source

Jul 26 11:42:48.971: MCAST-RWATCH: -af IPv4 -topo 80BC: Route watch topology found.
Jul 26 11:42:48.971: MCAST-RWATCH: -af IPv4 -topo 80BC -prefix 192.2.31.3/32: Route
watch prefix not found.
```

```
Jul 26 11:42:48.971: MCAST-RWATCH: -af IPv4 -topo 80BC -prefix 192.2.31.3/32: Route
watch prefix created.
Jul 26 11:42:48.971: MCAST-RWATCH: -af IPv4 -topo 80BC -prefix 192.2.31.3/32: Started
watching prefix in RIB.
Jul 26 11:42:48.971: MCAST-RWATCH: -af IPv4 -topo 80BC -prefix 192.2.31.3/32: Started
watching prefix in LISP.

// (S,G) entry created

Jul 26 11:42:48.972: MRT(188): WAVL Insert LISP interface: LISP0.4100 in
(192.2.31.3,239.100.100.100) Next-hop: 232.0.3.1 Outer-source: 192.2.101.66 Successful
Jul 26 11:42:48.972: MRT(188): Add LISP0.4100/232.0.3.1/192.2.101.66to the olist of
(192.2.31.3, 239.100.100.100), Forward state - MAC not built
Jul 26 11:42:49.106: MRT(188): (192.2.31.3,239.100.100.100), RPF change from  /0.0.0.0
to LISP0.4100/192.2.101.70
Jul 26 11:42:49.107: MRT(188): RPF change 192.2.101.70/LISP0.4100 for (192.2.31.3/32,
239.100.100.100)

// (S,G) S-bit PIM join sent to Edge1

Jul 26 11:42:49.107: PIM(188): Insert (192.2.31.3,239.100.100.100) join in nbr
192.2.101.70's queue
Jul 26 11:42:49.107: PIM(188): Building Join/Prune packet for nbr 192.2.101.70
Jul 26 11:42:49.107: PIM(188):  Adding v2 (192.2.31.3/32, 239.100.100.100), S-bit Join
Jul 26 11:42:49.107: PIM(188): Adding LISP Multicast transport attribute in join/prune
to 192.2.101.70 (LISP0.4100)
Jul 26 11:42:49.107: PIM(188): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```

You should also be able to see this (S,G) in the SA cache on Border2:

```
Border2#show ip msdp vrf Corp_VN sa-cache
MSDP Source-Active Cache - 1 entries
(192.2.31.3, 239.100.100.100), RP 192.2.199.7, BGP/AS 65003, 00:39:27/00:05:04, Peer
192.2.199.7
```

After this, the process is the same as what we've seen many times before in this document. Eventually, Edge2 will trigger a SPT cutover and that is how the end to end multicast tree will be built.

Let's quickly take a look at the final MRIB tables on all the devices:

```
Edge1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 01:01:51/stopped, RP 192.2.199.1, flags: SPF
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.65, LISP: [192.2.101.65, 232.0.3.1]
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 01:01:51/00:01:29, flags: FTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.1), Forward/Sparse, 01:01:50/stopped, Pkts:0, p
      192.2.101.71, 01:01:49/00:02:42
```


Aninda Chatterjee (aninchat@cisco.com)

In the underlay, for Edge1:

```
Edge1#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 01:05:20/00:03:02, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/4, Forward/Sparse, 01:05:20/00:03:02, Pkts:0, p
    GigabitEthernet1/0/2, Forward/Sparse, 01:05:20/00:02:58, Pkts:0, p

(192.2.101.65, 232.0.3.1), 01:05:20/00:00:38, flags: sT
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 192.2.101.68
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 01:05:20/stopped, Pkts:0
```

Border1, while it does have the entries for this group, it will do nothing with the packets because the outgoing interface is null.

```
Border1#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 01:01:50/stopped, RP 192.2.199.1, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(192.2.31.3, 239.100.100.100), 01:01:50/00:01:44, flags: PTA
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70, 232.0.3.1]
  Outgoing interface list: Null
```

Same thing in the underlay:

```
Border1#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 01:08:22/00:00:37, flags: sT
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 192.2.101.69
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 01:08:22/stopped

(192.2.101.65, 232.0.3.1), 01:08:23/00:03:01, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 01:08:23/00:03:01, p
```

Border2 also will not participate in the overlay because the packet is already encapsulated by the time it will reach it and it only needs to forward using the underlay SSM tree:

```
Border2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 01:13:04/00:02:43, RP 192.2.199.1, flags: Sp
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.66, 232.0.3.1), Forward/Sparse, 01:13:04/stopped, p
      192.2.101.71, 01:13:04/00:02:43

(192.2.31.3, 239.100.100.100), 01:01:50/00:02:17, flags: PMT
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70, 232.0.3.1]
  Outgoing interface list: Null
```

The underlay has the correct outgoing interface list:

```
Border2#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 01:11:25/00:02:55, flags: sTp
  Incoming interface: TenGigabitEthernet1/0/1, RPF nbr 192.2.101.77
  lisp eid refcnt: 1
  Outgoing interface list:
    TenGigabitEthernet1/0/4, Forward/Sparse, 01:11:24/00:02:55, p
    Null0, Forward/Dense, 01:11:25/stopped

(192.2.101.66, 232.0.3.1), 01:22:40/00:02:30, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/4, Forward/Sparse, 01:22:40/00:02:30, p
```

And finally, Edge2 will decapsulate the packet using the underlay and then forward towards the receiver using the lookup (post recirculation) in the overlay:

```
Edge2#show ip mroute 232.0.3.1 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.1), 01:13:35/00:01:24, flags: sT
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 192.2.101.74
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 01:13:35/stopped, Pkts:0

*snip*
```


Aninda Chatterjee (aninchat@cisco.com)

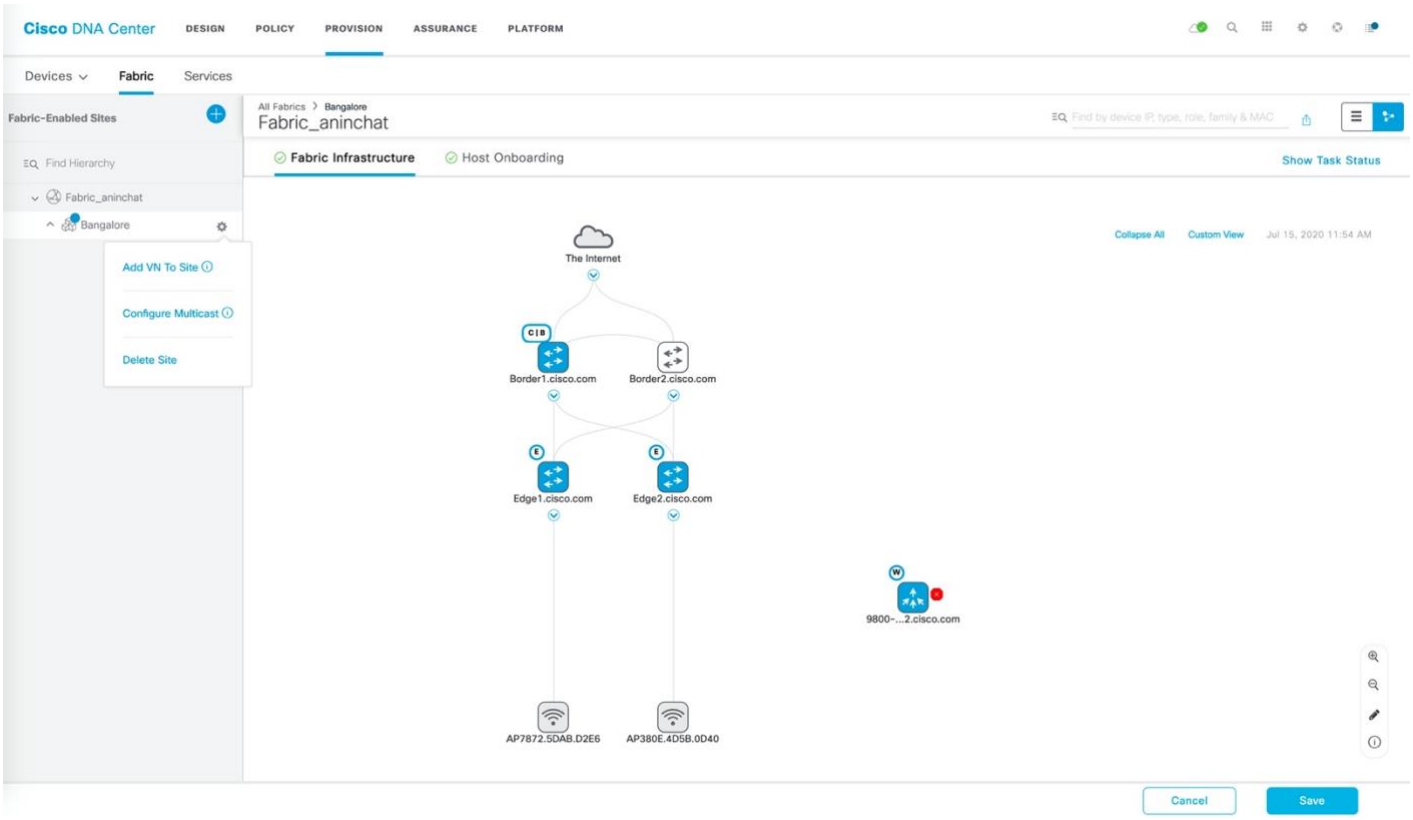
```
Edge2#show ip mroute vrf Corp_VN 239.100.100.100 verbose
IP Multicast Routing Table

*snip*

(*, 239.100.100.100), 01:13:04/stopped, RP 192.2.199.1, flags: SJC
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.66, LISP: [192.2.101.66, 232.0.3.1]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 01:13:04/00:02:42, Pkts:0

(192.2.31.3, 239.100.100.100), 01:01:49/00:01:20, flags: JT
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70, 232.0.3.1]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 01:01:49/00:02:42, Pkts:0
```


Aninda Chatterjee (aninchat@cisco.com)

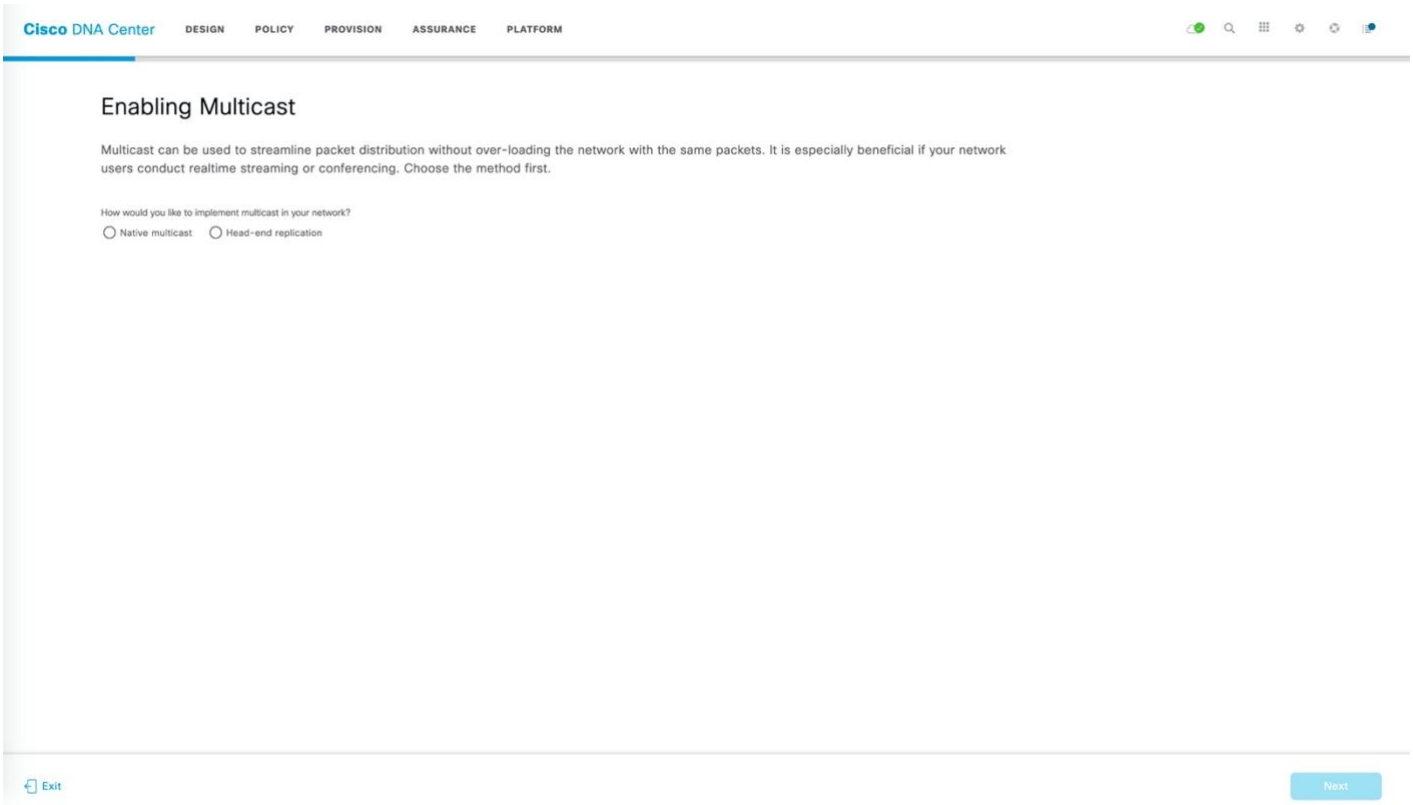
Native multicast – SSM

Configuring SSM in the overlay via DNAC

The process for this is similar to what we followed for ASM in the overlay. Traverse to your fabric page again and select ‘Configure Multicast’ once more:

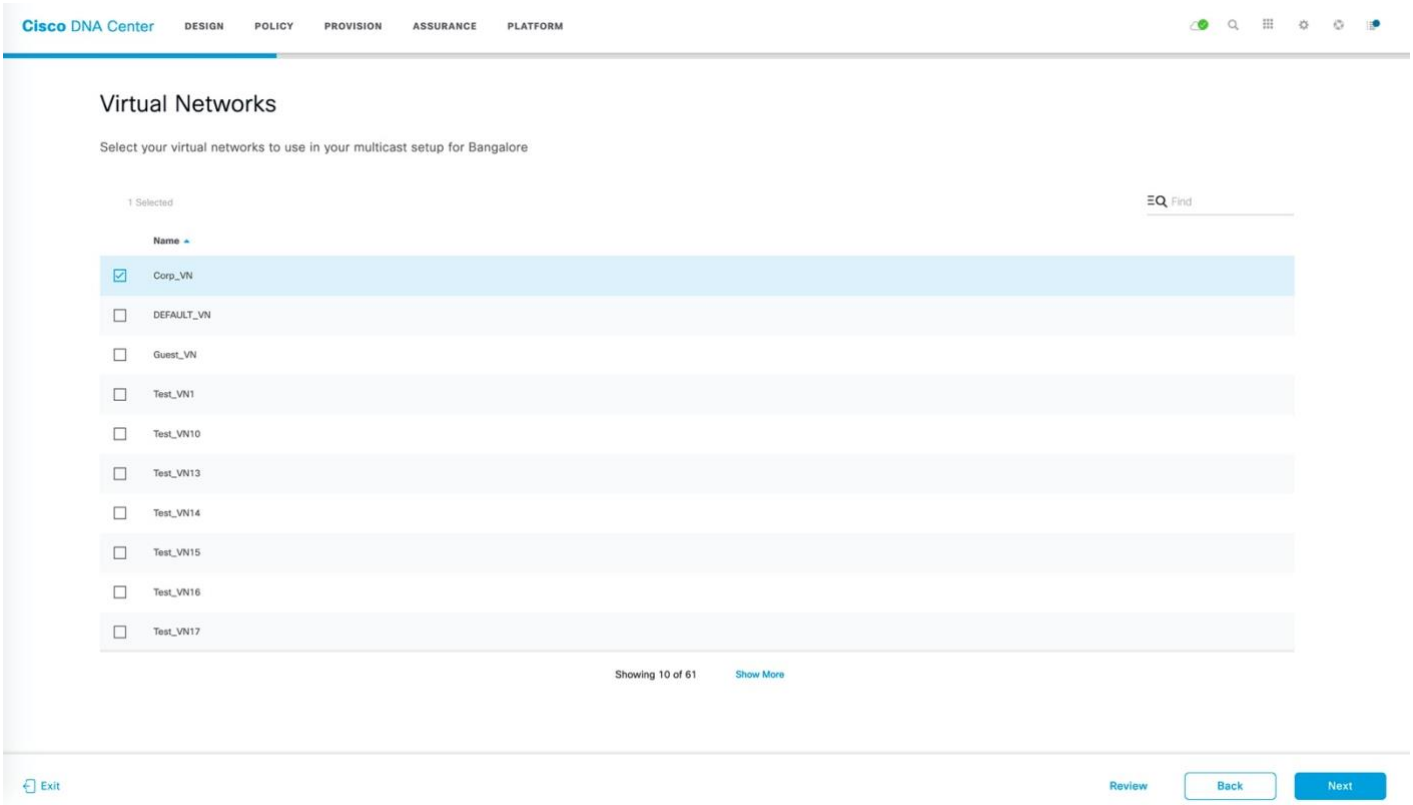


From here, choose native multicast:

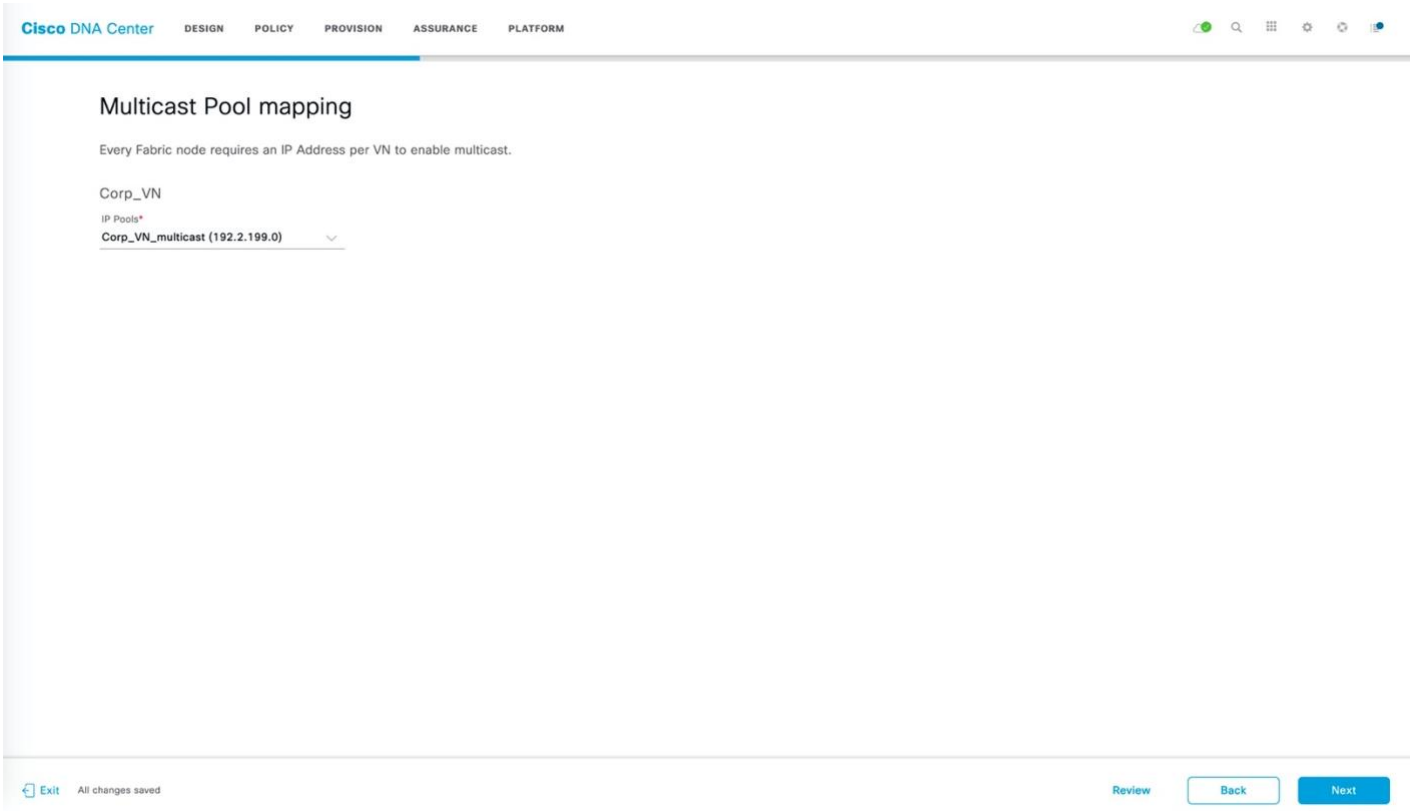


Aninda Chatterjee (aninchat@cisco.com)

Choose what VNs you want to enable this for. In our case, we want this for “Corp_VN”:

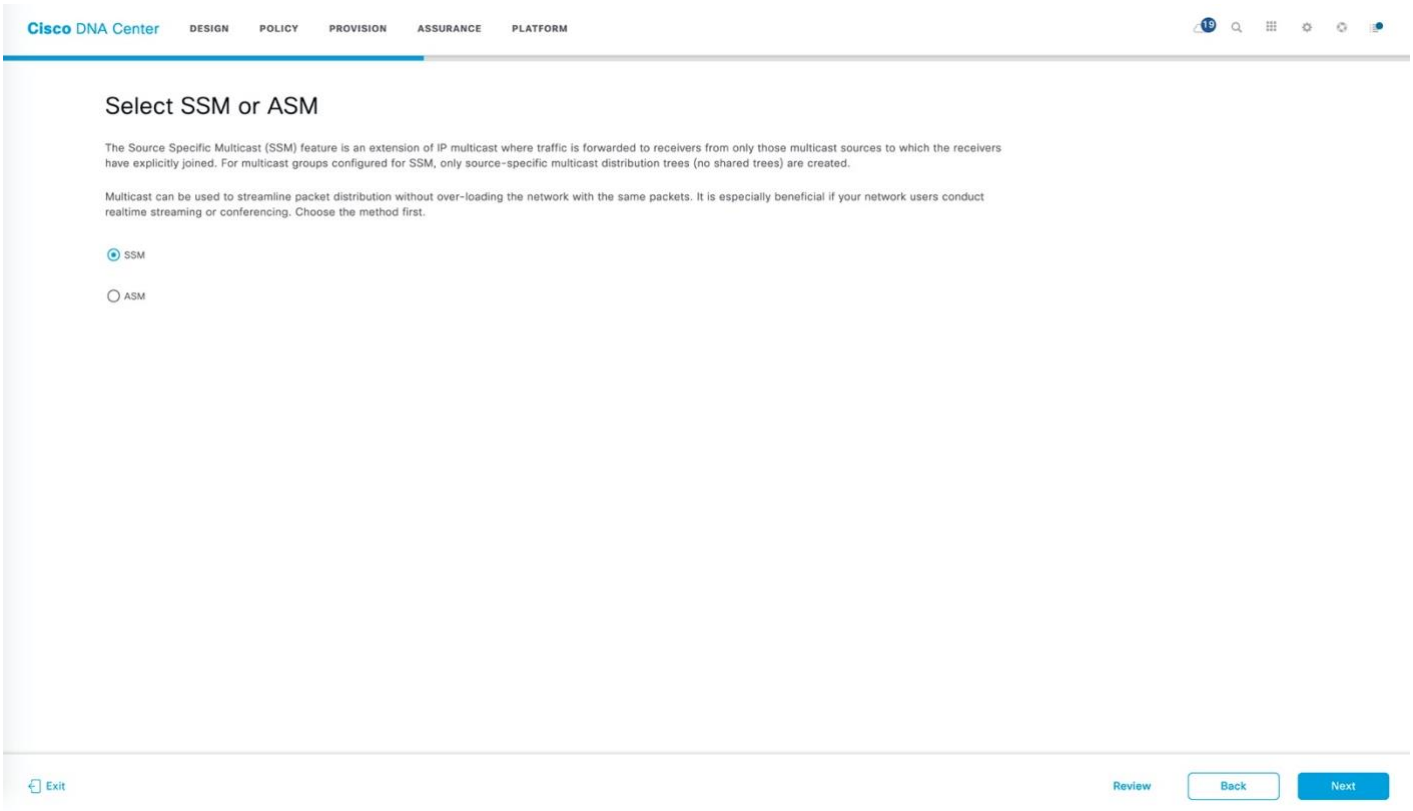


Next, choose a multicast pool:

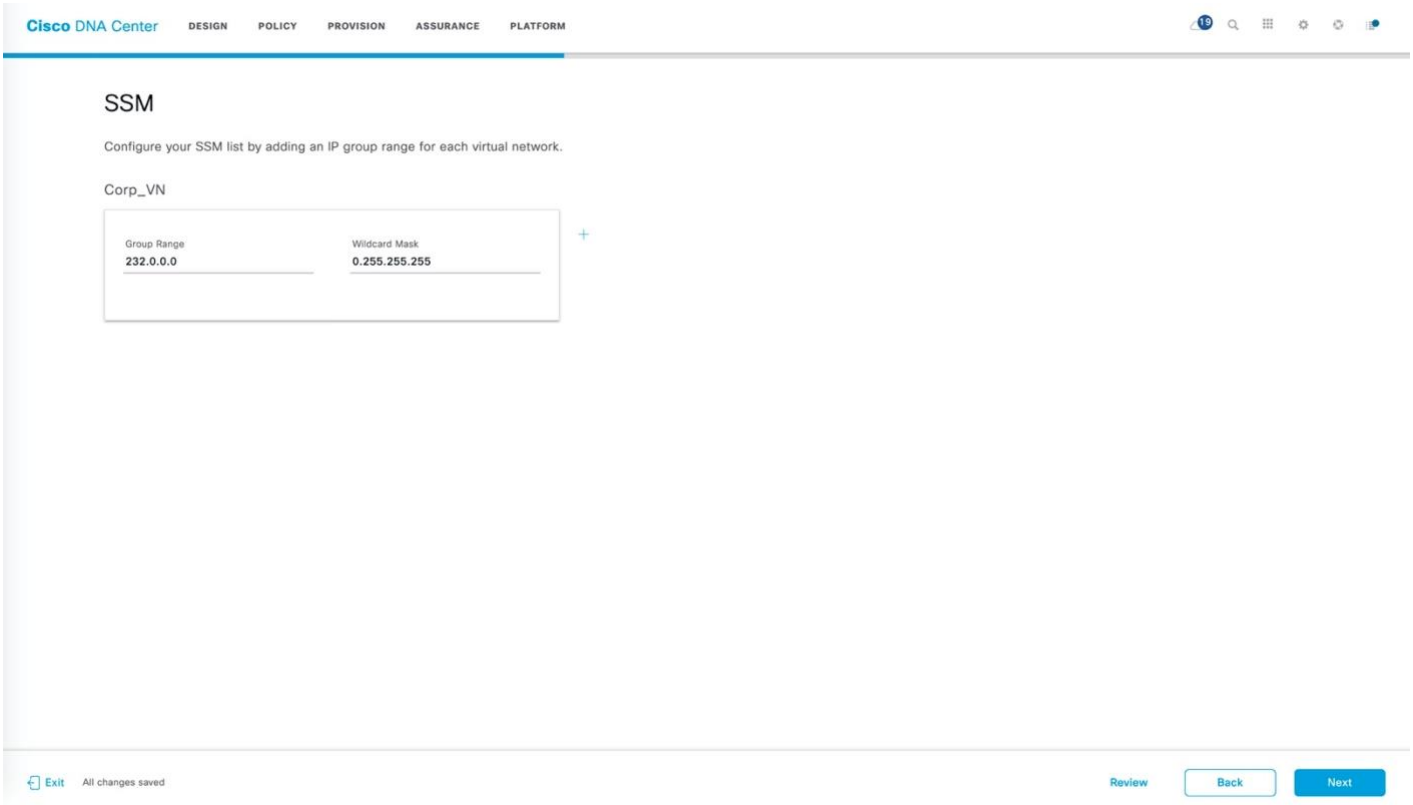


This pool must be pre-defined in DNAC and is used to create loopbacks on the fabric devices within the VRF. We will look at the exact configurations pushed shortly.

Next, choose whether you want to run ASM or SSM in the overlay. We will choose SSM:

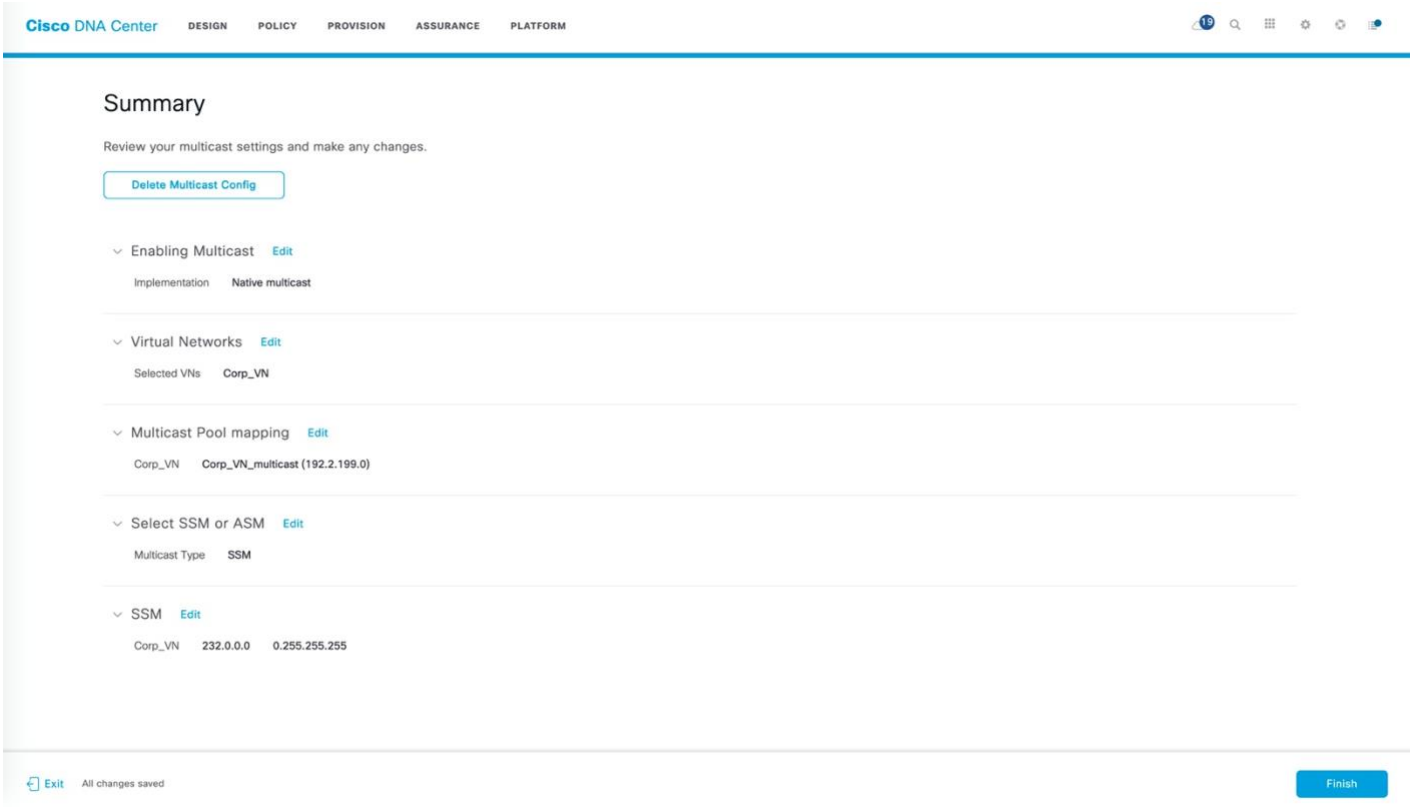


Define the SSM range that should be allowed in the overlay:



We have kept this as the default – 232.0.0.0/8. Next, you should get a final summary of your choices:

Aninda Chatterjee (aninchat@cisco.com)



Once you click “Finish”, DNAC will push the appropriate configuration to the fabric devices.

The following configuration is pushed per device in our topology:

Edge1:

```
// ACL to control SSM allowed range

006313: Jul 20 17:33:08.594: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:!exec: enable
006314: Jul 20 17:33:10.055: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip access-list standard SSM_RANGE_Corp_VN
006315: Jul 20 17:33:10.076: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:permit 232.0.0.0 0.255.255.255

// enable multicast routing for Corp_VN VRF

006387: Jul 20 17:33:12.596: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast-routing vrf Corp_VN

// enable multicast transport under LISP interface for this VRF
// define underlay SSM range that will be used

006388: Jul 20 17:33:12.614: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface LISP0.4100
006389: Jul 20 17:33:12.640: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp transport multicast
006390: Jul 20 17:33:12.654: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp core-group-range 232.0.0.1 1000
```

```
// enabling IGMPv3 on host facing SVIs

006391: Jul 20 17:33:12.670: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan1029
006392: Jul 20 17:33:12.697: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp explicit-tracking
006393: Jul 20 17:33:12.712: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim passive
006394: Jul 20 17:33:12.732: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip igmp version 3
006395: Jul 20 17:33:12.740: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:exit

// loopback4100 created and configured in the VRF and enabled with PIM-SM

006406: Jul 20 17:33:12.872: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4100
006407: Jul 20 17:33:12.888: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
006408: Jul 20 17:33:12.907: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.2 255.255.255.255
006409: Jul 20 17:33:12.915: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// PIM SSM enabled and ACL applied to allow defined range only

006410: Jul 20 17:33:12.948: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN register-source Loopback4100
006411: Jul 20 17:33:12.982: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN ssm range SSM_RANGE_Corp_VN

// loopback4100s IP mapped under LISP database

006433: Jul 20 17:33:13.657: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:instance-id 4100
006434: Jul 20 17:33:13.666: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:service ipv4
006439: Jul 20 17:33:13.821: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:database-mapping 192.2.199.2/32  locator-set rloc_70ac0729-86f6-44d0-a1c9-
254b98507c24
```

The configuration for Edge2 will be similar so we'll not cover that.

Border1:

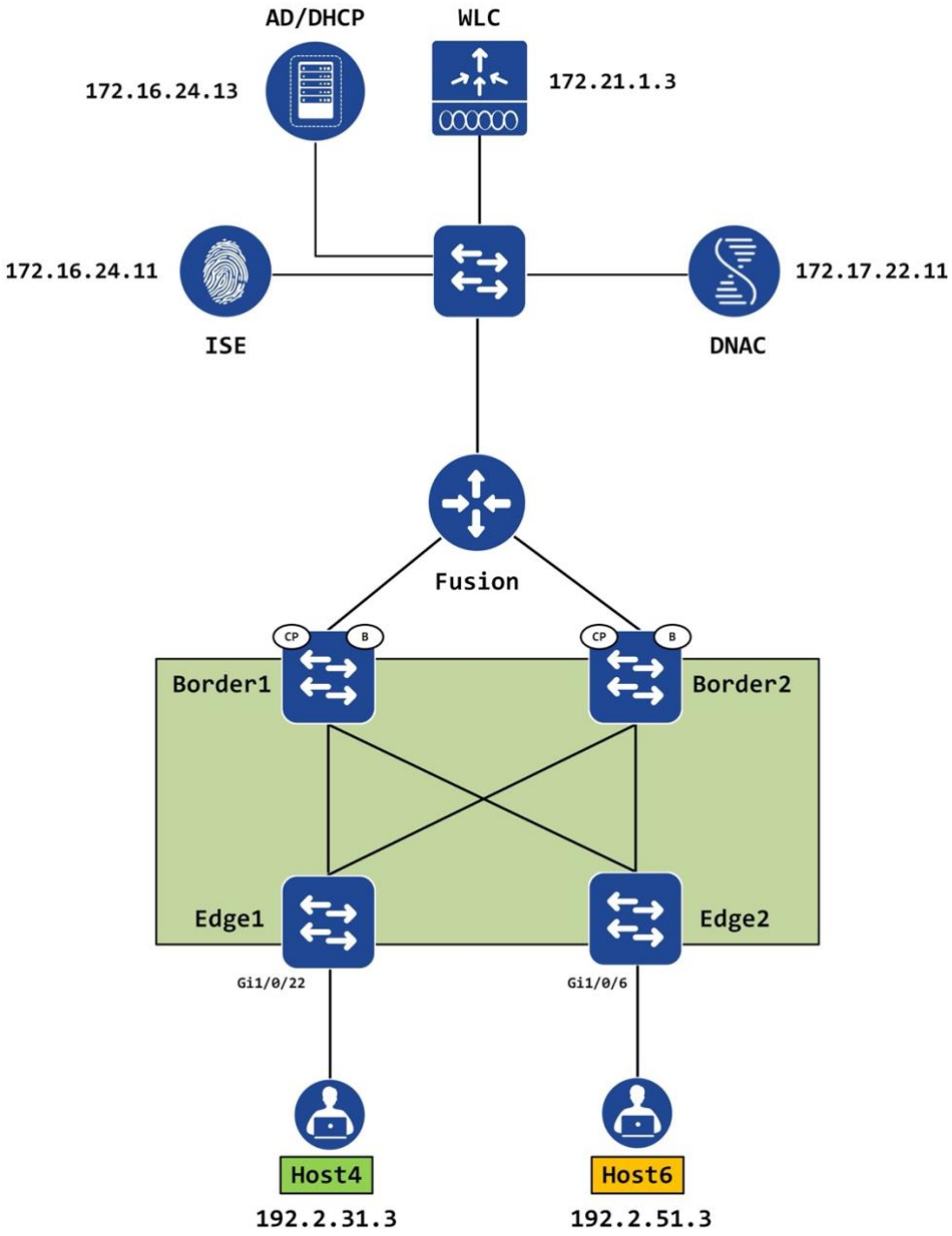
```
// ACL defined for allowed SSM range

Jul 20 2020 17:32:38.167 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip access-list standard SSM_RANGE_Corp_VN
Jul 20 2020 17:32:38.218 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:permit 232.0.0.0 0.255.255.255
Jul 20 2020 17:32:38.264 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip multicast-routing vrf Corp_VN
```

```
// LISP interface enabled for multicast transport and underlay SSM range defined

Jul 20 2020 17:32:38.299 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface LISP0.4100
Jul 20 2020 17:32:38.359 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp transport multicast
Jul 20 2020 17:32:38.430 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim lisp core-group-range 232.0.0.1 1000

// L3 handoff interface (for corresponding VRF) enabled with PIM

Jul 20 2020 17:32:38.460 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Vlan3014
Jul 20 2020 17:32:38.488 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode
Jul 20 2020 17:32:38.519 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:exit

// loopback4100 created and enabled with PIM

Jul 20 2020 17:32:38.554 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:interface Loopback4100
Jul 20 2020 17:32:38.595 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:vrf forwarding Corp_VN
Jul 20 2020 17:32:38.663 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip address 192.2.199.4 255.255.255.255
Jul 20 2020 17:32:38.697 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim sparse-mode

// PIM SSM enabled with ACL for defined overlay SSM range

Jul 20 2020 17:32:38.768 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN register-source Loopback4100
Jul 20 2020 17:32:38.837 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:ip pim vrf Corp_VN ssm range SSM_RANGE_Corp_VN

// multicast pool used to create loopbacks advertised in BGP

Jul 20 2020 17:32:38.884 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:router bgp 65003
Jul 20 2020 17:32:38.925 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:address-family ipv4 vrf Corp_VN
Jul 20 2020 17:32:38.991 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:aggregate-address 192.2.199.0 255.255.255.0 summary-only

// loopback4100s IP added in LISP database

Jul 20 2020 17:32:39.646 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:instance-id 4100
Jul 20 2020 17:32:39.663 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:service ipv4
Jul 20 2020 17:32:39.700 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:route-export site-registrations
Jul 20 2020 17:32:39.743 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:distance site-registrations 250
Jul 20 2020 17:32:39.771 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:map-cache site-registration
```

```
Jul 20 2020 17:32:39.902 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:database-mapping 192.2.199.4/32  locator-set rloc_23bd3133-ba7b-4058-9552-
e37e69cbc843

// multicast pool range added in LISP site configuration

Jul 20 2020 17:32:48.699 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:site site_uci
Jul 20 2020 17:32:48.772 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:authentication-key uci
Jul 20 2020 17:32:48.873 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:aninchat  logged
command:eid-record instance-id 4100 192.2.199.0/24  accept-more-specifics
```

SSM in SDA – source and receiver in fabric

In this case, we have a new host (Host6) that is connected to Edge2 (authenticated and authorized to the same subnet as Host) that will act as a receiver that is interested in multicast data sent to 232.50.50.50, specifically from Host4 (192.2.31.3).



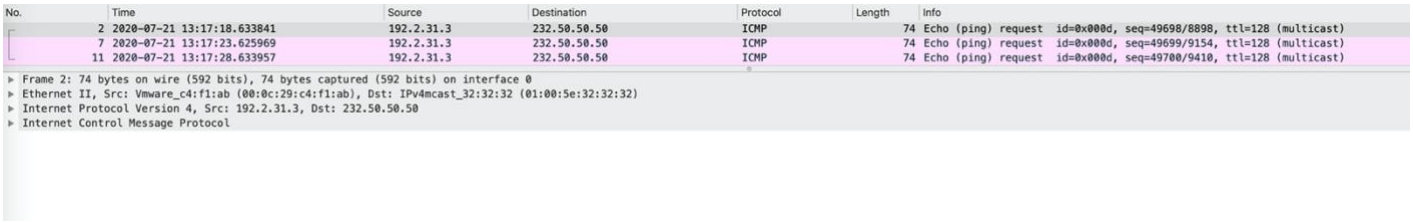
Aninda Chatterjee (aninchat@cisco.com)

Start of source stream

The source, Host4, starts to send traffic to 232.50.50.50. However, we do not see any entries for this group in the multicast routing table for the VRF:

```
Edge1#show ip mroute vrf Corp_VN
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 02:14:30/00:02:24, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback4100, Forward/Sparse, 02:14:30/00:02:24
```

Is the source really sending traffic? Let’s take a quick capture on the source facing interface to confirm:



Okay, so we do see traffic coming in but there’s still no state built for it in the MRIB. Why? Remember, this is SSM – the receiver and the last-hop router drive the multicast tree. According to the SSM RFC (RFC 4607):

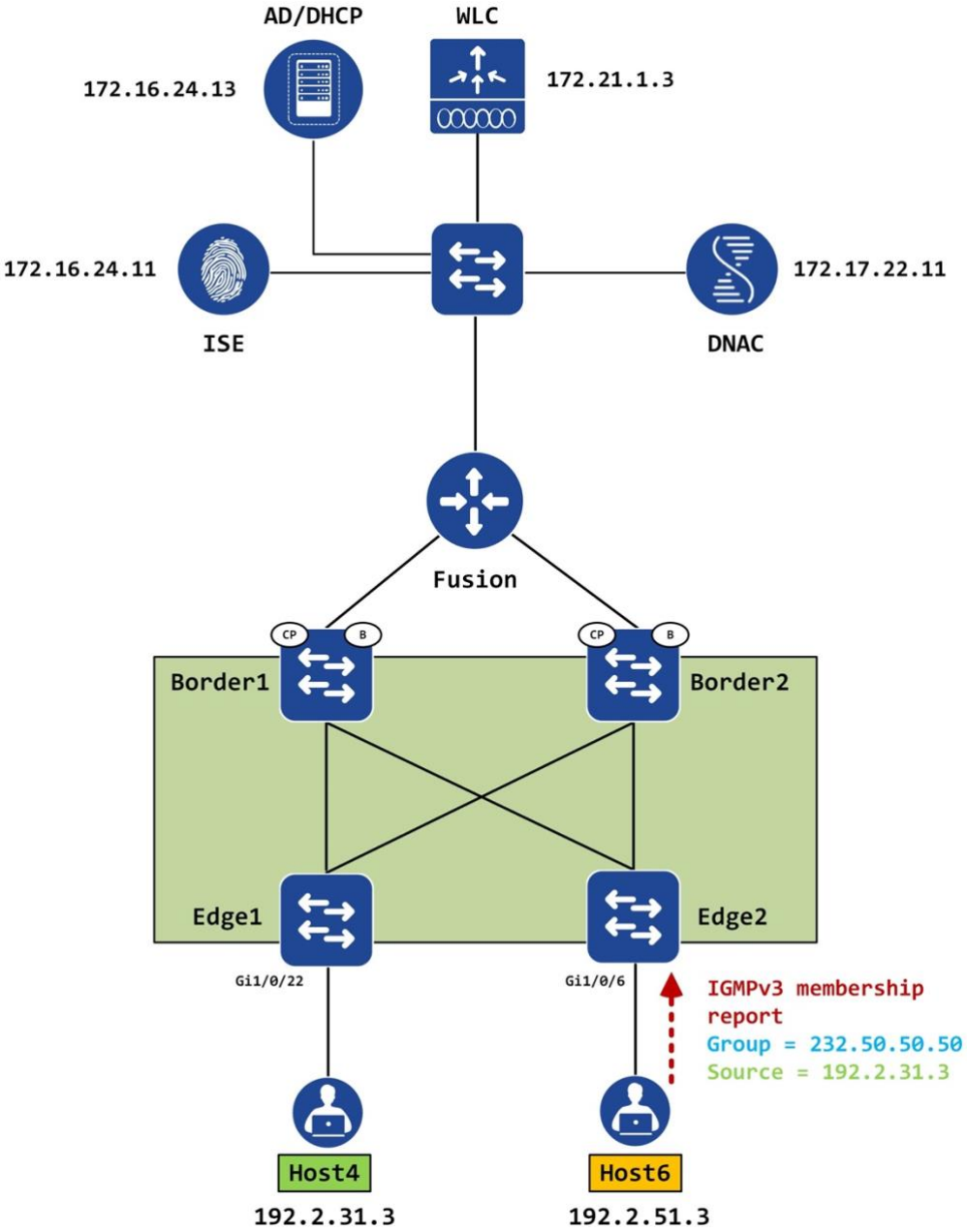
“5.1. Packet Forwarding

> A router that receives an IP datagram with a source-specific destination address MUST silently drop it unless a neighboring host or router has communicated a desire to receive packets sent from the source and to the destination address of the received packet.”

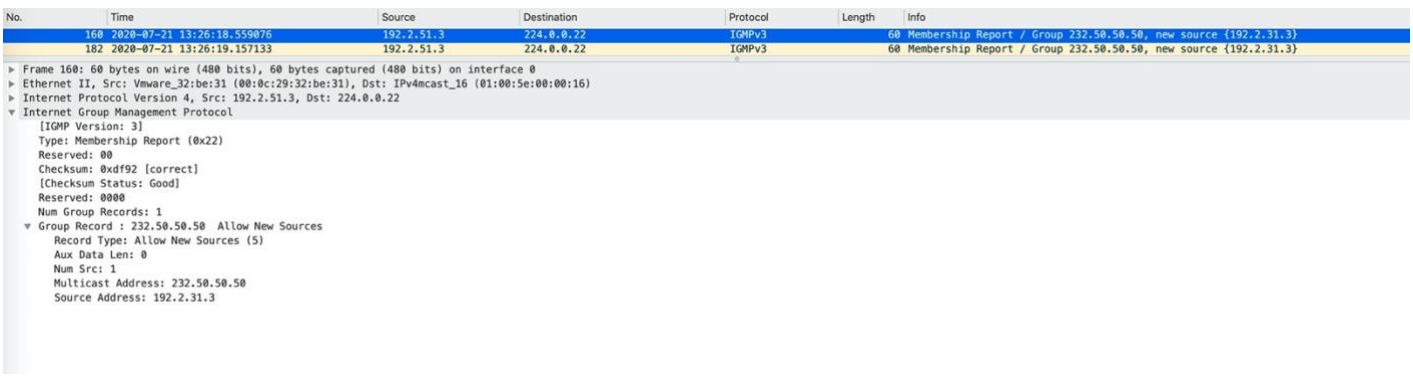
Thus, Edge1 is simply dropping the datagram because there is no receiver that is interested in it yet.

Receiver join and building the SPT

Host6 sends an IGMPv3 membership report to join the group 232.50.50.50 and the source 192.2.31.3.



A packet capture also confirms this:



Aninda Chatterjee (aninchat@cisco.com)

We can additionally confirm that this was installed as a v3 join in the IGMP table as well:

```
Edge2#show ip igmp vrf Corp_VN membership
Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP, the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude

 Channel/Group                  Reporter        Uptime   Exp.  Flags  Interface
 *,239.255.255.250              0/1             20:14:38 02:21 3AT    Vl1029
/*,232.50.50.50                 1/0             00:02:37 stop  3MAT   Vl1030
 192.2.31.3,232.50.50.50        1/0             00:02:37 02:06 T      Vl1030
 *,224.0.1.40                   192.2.199.3     20:14:38 02:27 2LA    Lo4100
```

This IGMP report now kick starts the entire SSM process. A (S,G) entry is creating in the multicast routing table and like we saw in the ASM mode (for the underlay SSM), route watch gets triggered to resolve the source and determine the RLOC for it.

Once that is done, a PIM (S,G) join is sent towards this RLOC.

Let’s look at the debugs on Edge2 for this:

```
// route watch triggered for source

176311: Jul 21 12:10:59.632: MCAST-RWATCH: -af IPv4 -topo 8002: Route watch topology
not found.
176312: Jul 21 12:10:59.632: MCAST-RWATCH: -af IPv4 -topo 8002: Route watch topology
created.
176313: Jul 21 12:10:59.632: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix not found.
176314: Jul 21 12:10:59.632: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Route watch prefix created.
176315: Jul 21 12:10:59.632: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in RIB.
176316: Jul 21 12:10:59.632: MRT(2): RPF Track start on 192.2.31.3 for (192.2.31.3,
232.50.50.50)

// create (S,G) entry

176333: Jul 21 12:10:59.633: MRT(2): Create (192.2.31.3,232.50.50.50), RPF (Vlan1029,
0.0.0.0, 0/0)
176339: Jul 21 12:10:59.633: MRT(2): Set the T-flag for (192.2.31.3, 232.50.50.50)
176349: Jul 21 12:10:59.634: MRT(2): Add Vlan1030/232.50.50.50 to the olist of
(192.2.31.3, 232.50.50.50), Forward state - MAC not built
```


Aninda Chatterjee (aninchat@cisco.com)

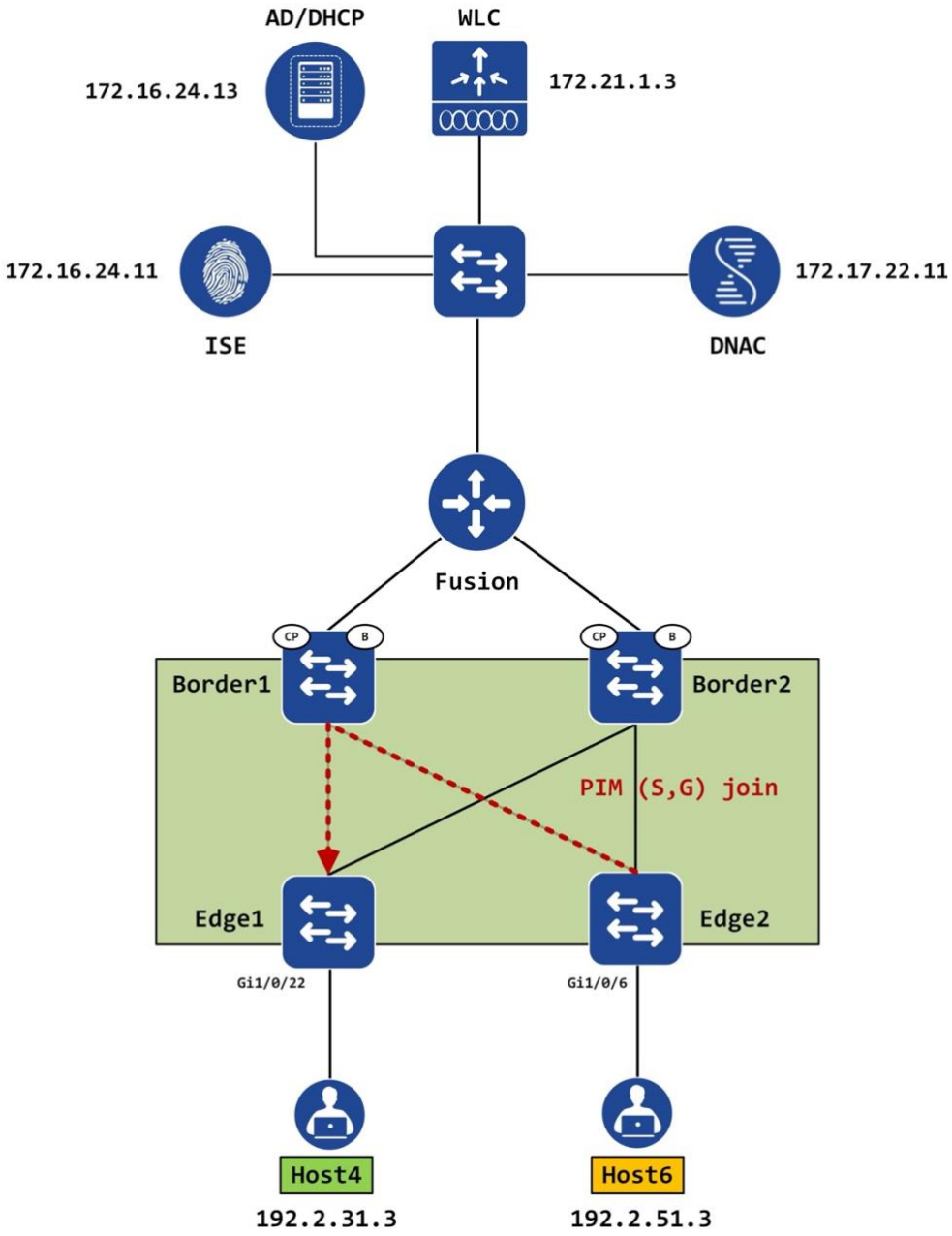
```
// route watch resolved source

176400: Jul 21 12:10:59.640: MCAST-RWATCH: Bound to LISP.
176401: Jul 21 12:10:59.640: MCAST-RWATCH: -af IPv4 -topo 8002: Route watch topology
found.
176402: Jul 21 12:10:59.640: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Started watching prefix in LISP.
176403: Jul 21 12:10:59.641: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Received LISP notification, masklen = 32.
176404: Jul 21 12:10:59.641: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Prefix status set to: LISP.
176405: Jul 21 12:10:59.641: MCAST-RWATCH: -af IPv4 -topo 8002 -prefix 192.2.31.3/32:
Prefix enqueued for notification.
176409: Jul 21 12:10:59.641: MCAST-RWATCH: -af IPv4 -topo 8000: Route watch topology
found.
176410: Jul 21 12:10:59.641: MCAST-RWATCH: -af IPv4 -topo 8000 -prefix 192.2.101.70/32:
Route watch prefix found.

// RPF updated for (S,G)

176412: Jul 21 12:10:59.641: MRT(2): (192.2.31.3,232.50.50.50), RPF change from
Vlan1029/0.0.0.0 to LISP0.4100/192.2.101.70
176419: Jul 21 12:10:59.642: MRT(2): RPF change 192.2.101.70/LISP0.4100 for
(192.2.31.3/32, 232.50.50.50)
```

Once this (S,G) is built, a S-bit join is triggered to 192.2.101.70 (the RLOC IP that was resolved for the source, 192.2.31.3).

```
176467: Jul 21 12:11:00.472: PIM(2): Insert (192.2.31.3,232.50.50.50) join in nbr
192.2.101.70's queue
176468: Jul 21 12:11:00.472: MRT(2): Start 0x7F8195C83188 [1787] send join timer
(192.2.31.3, 232.50.50.50) delay 58 sec , rpf_nbr 192.2.101.70/0.0.0.0
176469: Jul 21 12:11:00.472: PIM(2): Building Join/Prune packet for nbr 192.2.101.70
176470: Jul 21 12:11:00.473: PIM(2):  Adding v2 (192.2.31.3/32, 232.50.50.50), S-bit
Join
176471: Jul 21 12:11:00.473: PIM(2): Adding LISP Multicast transport attribute in
join/prune to 192.2.101.70 (LISP0.4100)
176472: Jul 21 12:11:00.473: PIM(2): Send v2 join/prune to 192.2.101.70 (LISP0.4100)
```

This process can be visualized like so:

Aninda Chatterjee (aninchat@cisco.com)



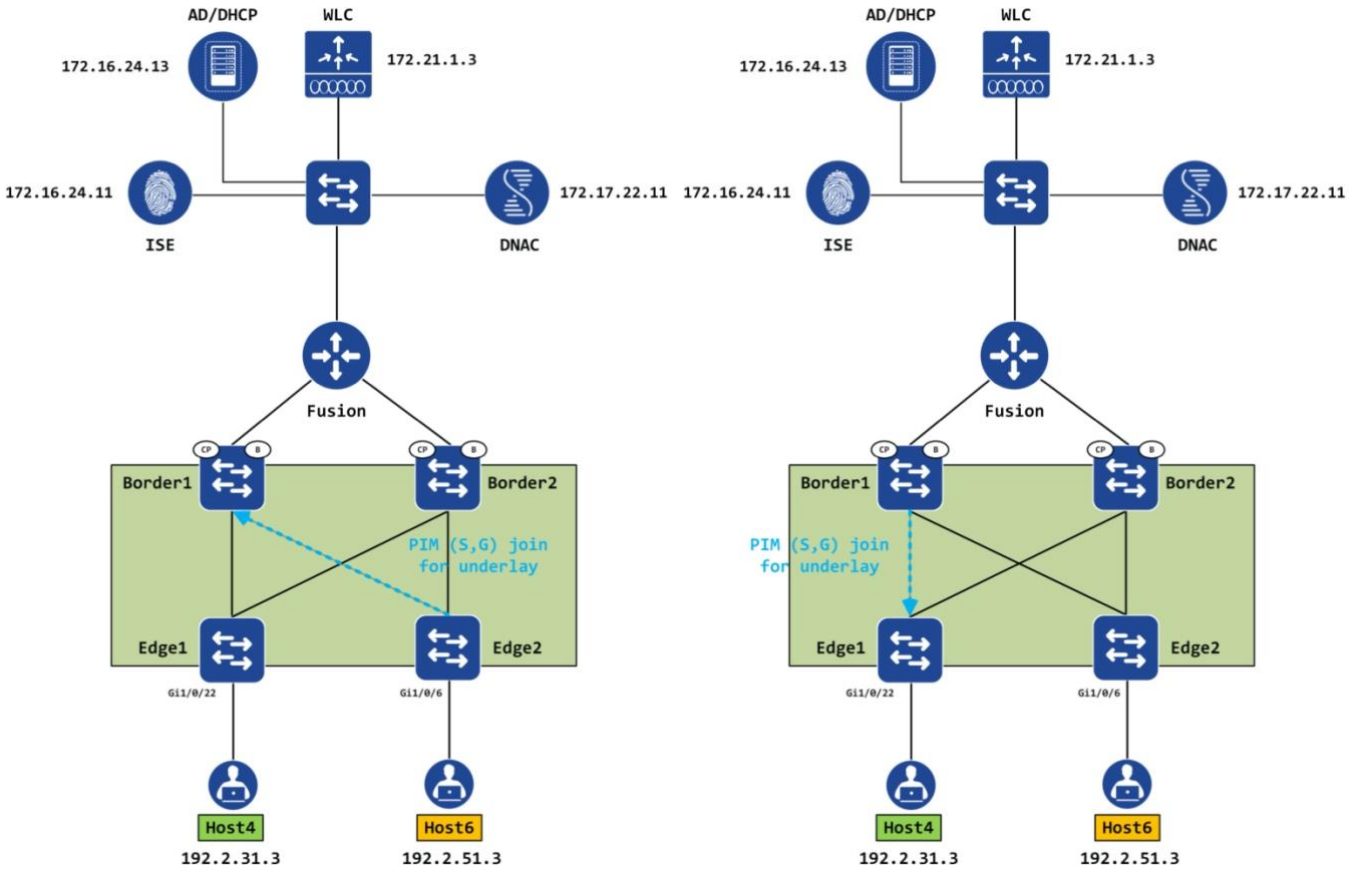
Remember, this is still native multicast! Which implies that an underlay SSM group must also be derived for the overlay SSM group. Like before, the simplest way to look at the underlay group derived is the by adding the “verbose” option to the multicast table CLI.

This allows you to quickly see what the underlay group is and this should be the same group derived on all your fabric devices.

Let’s take a look at this now:

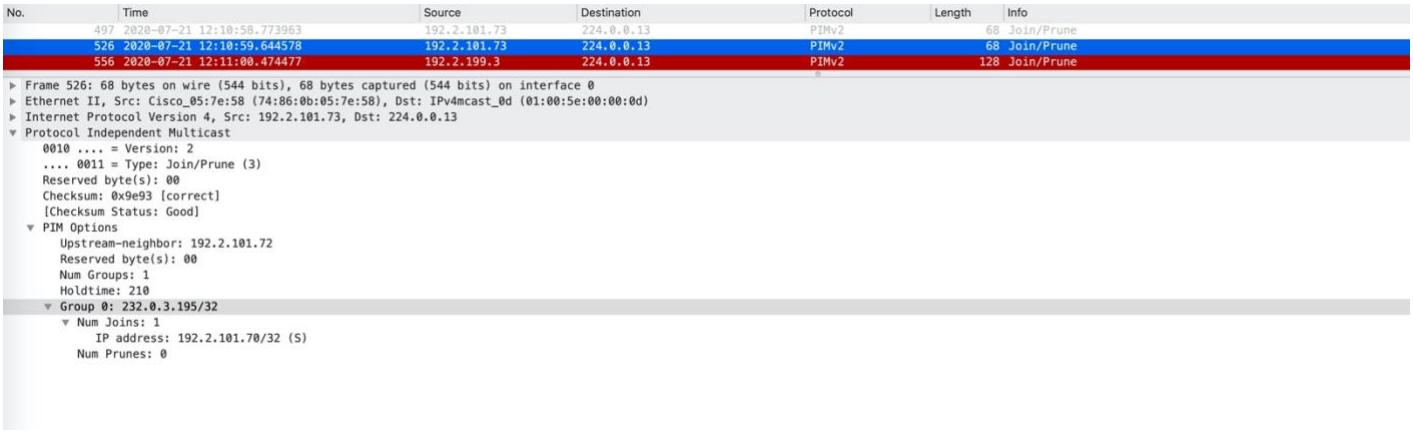
```
Edge2#show ip mroute vrf Corp_VN 232.50.50.50 verbose
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.31.3, 232.50.50.50), 00:48:48/00:02:49, flags: sTI
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70,
232.0.3.195]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 00:48:48/00:02:49
```

As can be seen from the above, the underlay SSM group derived is 232.0.3.195. The route watch process has already resolved the source to 192.2.101.70 and for the underlay also, a PIM (S,G) join will be sent to 192.2.101.70. A (S,G) state will be built hop by hop on any intermediate devices (Border1, for example, in this case).

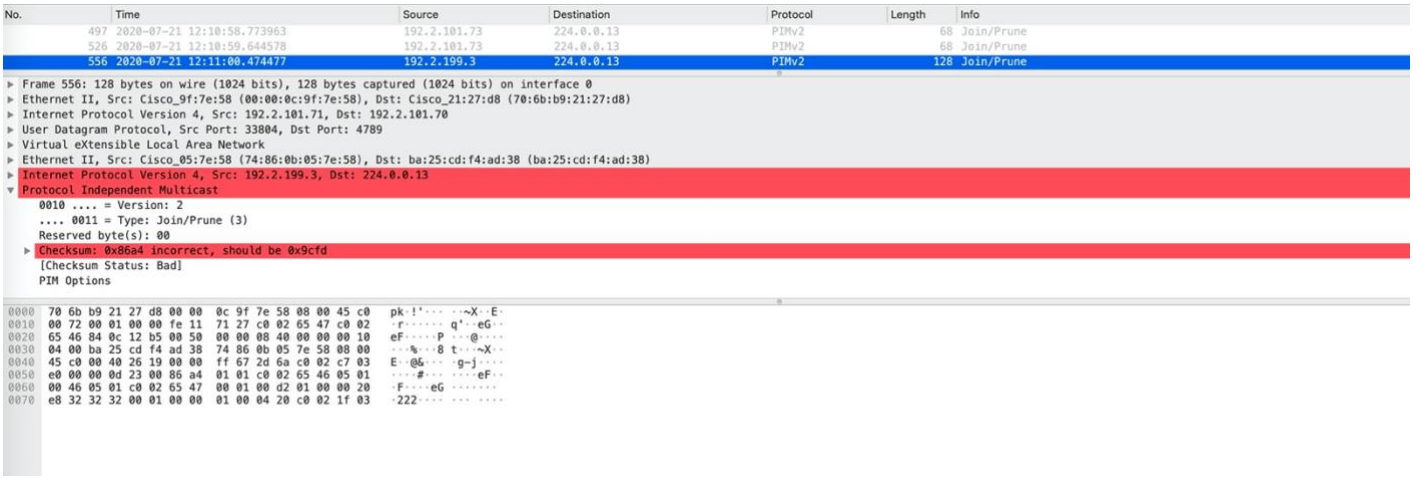


Aninda Chatterjee (aninchat@cisco.com)

Both of the overlay and underlay PIM (S,G) joins can be captured. The underlay SSM join:



The join is for group 232.0.3.195 and for a source of 192.2.101.70 (which is Edge1). The overlay join is captured below:



Like before, we need to read the hex dump to understand the PIM options.

```
0000   70 6b b9 21 27 d8 00 00 0c 9f 7e 58 08 00 45 c0
0010   00 72 00 01 00 00 fe 11 71 27 c0 02 65 47 c0 02
0020   65 46 84 0c 12 b5 00 50 00 00 08 40 00 00 00 10
0030   04 00 ba 25 cd f4 ad 38 74 86 0b 05 7e 58 08 00
0040   45 c0 00 40 26 19 00 00 ff 67 2d 6a c0 02 c7 03
0050   e0 00 00 0d 23 00 86 a4 01 01 c0 02 65 46 05 01
0060   00 46 05 01 c0 02 65 47 00 01 00 d2 01 00 00 20
0070   e8 32 32 32 00 01 00 00 01 00 04 20 c0 02 1f 03
```

We’re not going to go into detail of how to read this again – however, the important part is highlighted above. ‘05 01 00’ gives us a transport type of multicast which confirms that this is native multicast indeed.

Final MRIB state and packet flow

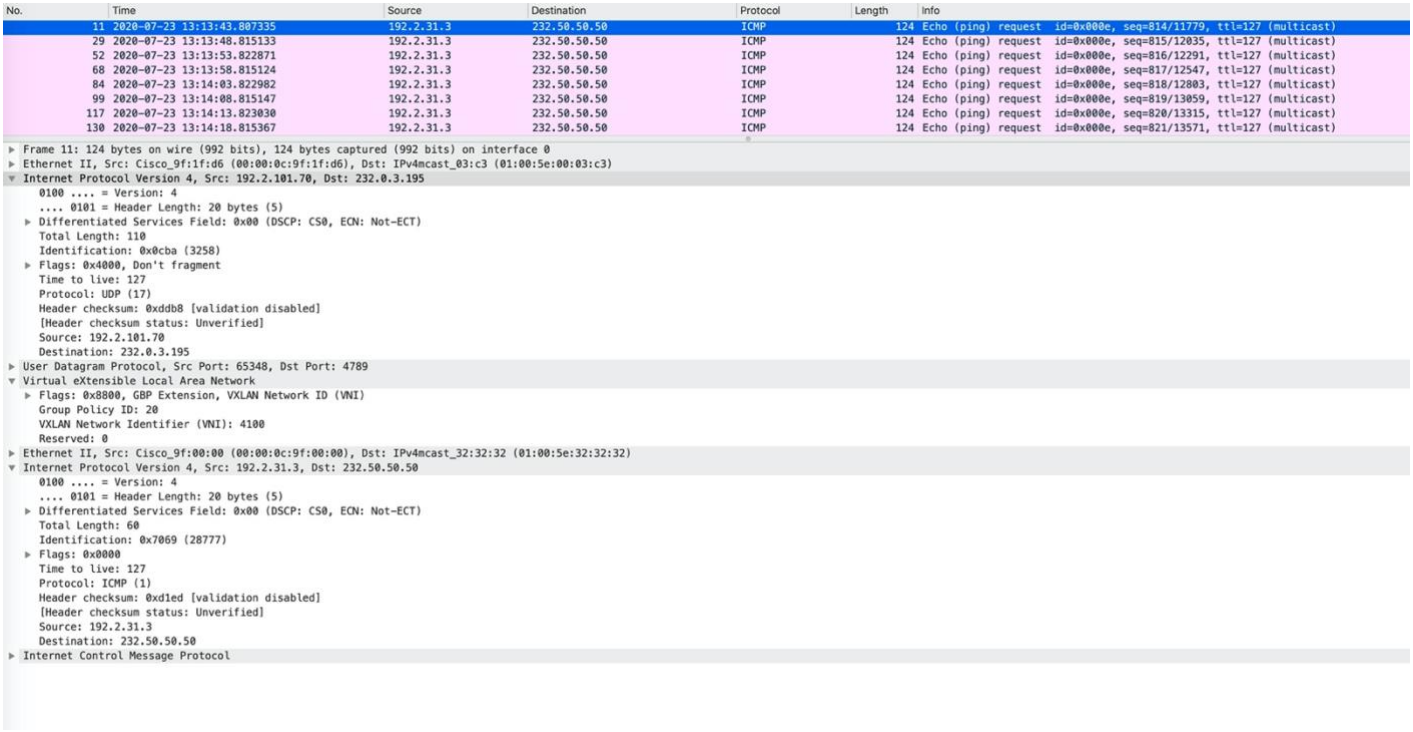
Edge1 has the following state now:

```
Edge1#show ip mroute vrf Corp_VN 232.50.50.50 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.31.3, 232.50.50.50), 00:50:03/00:03:21, flags: sTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.195), Forward/Sparse, 00:50:03/stopped, Pkts:0,
p
      192.2.101.71, 00:49:21/00:03:21
```

The underlay multicast routing table:

```
Edge1#show ip mroute 232.0.3.195 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.70, 232.0.3.195), 00:53:25/00:03:18, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:53:25/00:03:18, Pkts:0, p
```

The packet from the source should be encapsulated with the underlay SSM group that was derived. Let's take a packet capture to confirm:



The packet shows the correct encapsulation. Once Edge1 encapsulates and forwards it out, the multicast routing is now done purely on the outer header, which maps to the underlay SSM tree. This comes to Border1, which has a final state of:

```
Border1#show ip mroute vrf Corp_VN 232.50.50.50 verbose
Group 232.50.50.50 not found
```

This is expected because in the overlay, the RP is pruned since it is not a part of the SPT (since it forwards only on the outer multicast destination). Instead, confirm the underlay multicast tree:

```
Border1#show ip mroute 232.0.3.195 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.195), 01:04:59/00:03:05, flags: sTp
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 192.2.101.69
  Outgoing interface list:
    GigabitEthernet1/0/3, Forward/Sparse, 01:04:18/00:03:05, p
```

Finally, this reaches Edge2. Again, since the outer header is the underlay SSM tree, we must check that first:

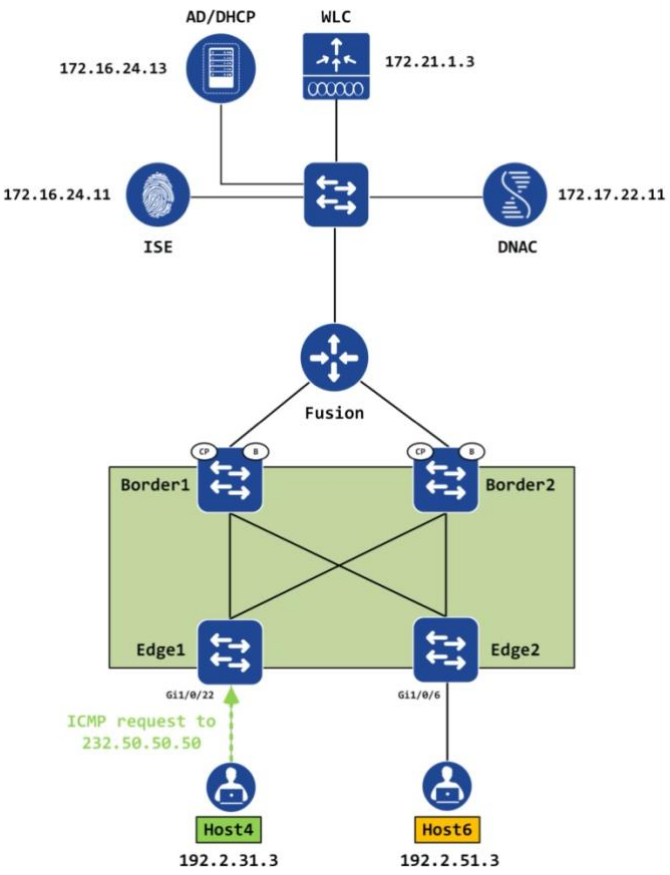
```
Edge2#show ip mroute 232.0.3.195
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join 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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.70, 232.0.3.195), 01:06:12/00:02:46, flags: sT
  Incoming interface: GigabitEthernet1/0/3, RPF nbr 192.2.101.72
  Outgoing interface list:
    Null0, Forward/Dense, 01:06:12/stopped
```

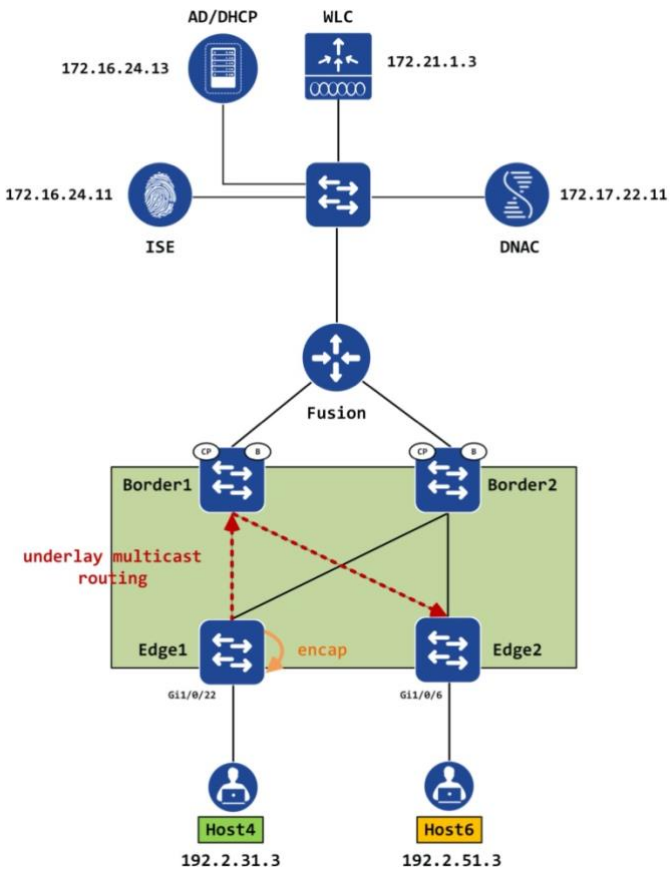
An outgoing interface of Null0 implies a packet decapsulation here. Once decapsulated, it is recirculated for a lookup again, which will now happen in the VRF specific multicast table:

```
Edge2#show ip mroute vrf Corp_VN 232.50.50.50 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.31.3, 232.50.50.50), 19:39:37/00:02:21, flags: sTI
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70,
232.0.3.195]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 19:39:37/00:02:21, Pkts:0
```

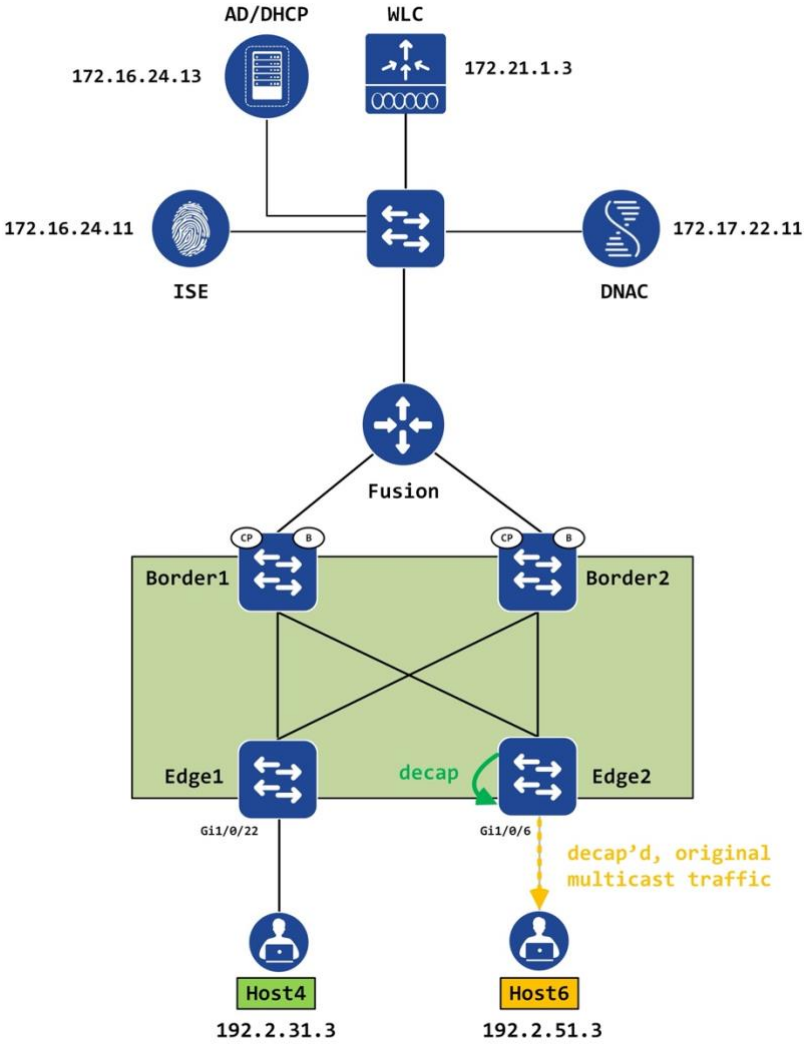

Aninda Chatterjee (aninchat@cisco.com)

Visually, this flow would be:





Aninda Chatterjee (aninchat@cisco.com)



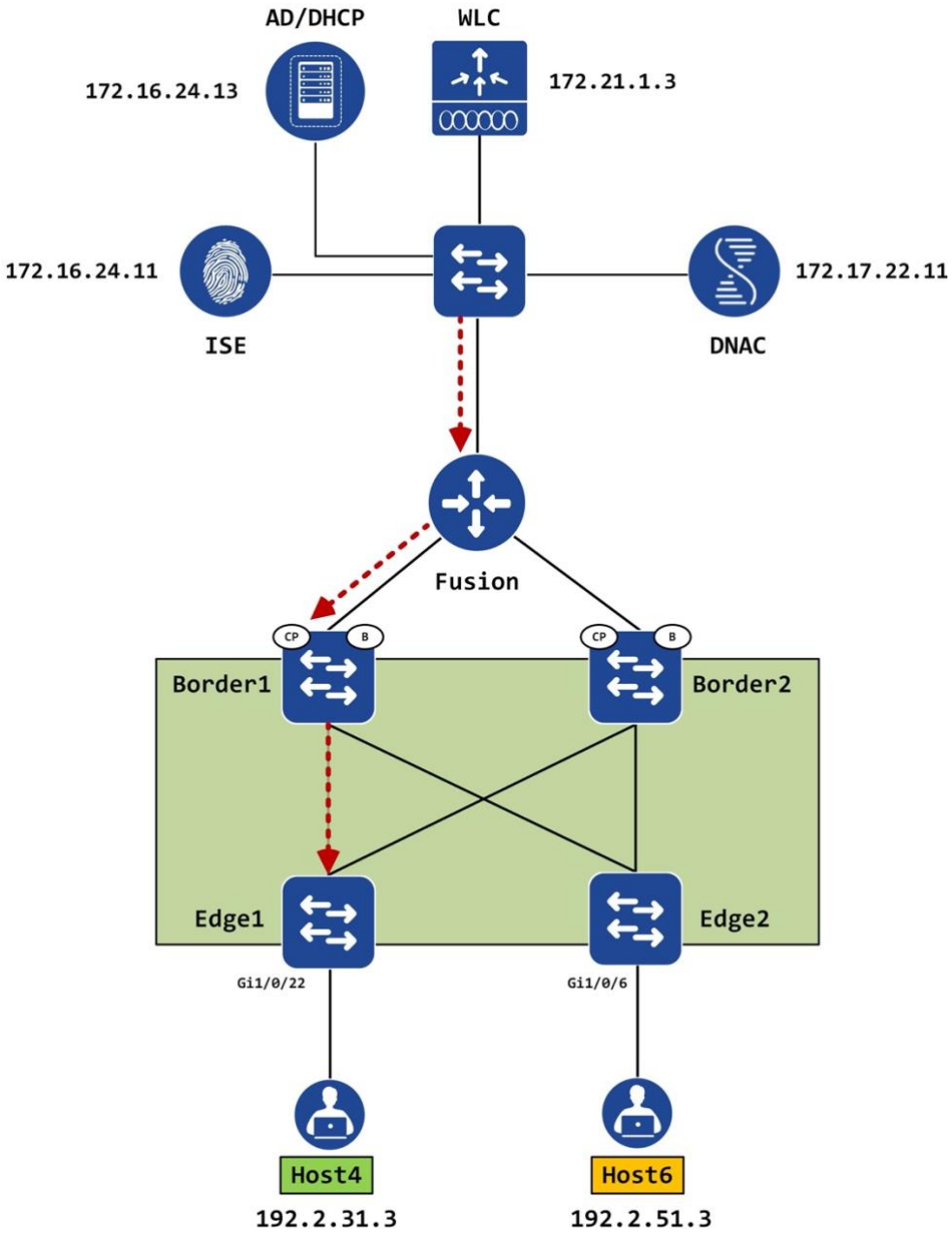
Aninda Chatterjee (aninchat@cisco.com)

SSM in SDA – source in fabric and receiver outside fabric

Receiver join and building SPT

The general workflows are quite similar so we're not going to go too deep with debugs. The only difference here is that the receiver is outside the fabric, making the RP (Border1) a part of the SPT and thus a part of the overlay tree as well.

The tree is now built from the shared services switch and the (S,G) is sent hop by hop with every intermediate switch building the (S,G) state.



Final MRIB state and packet flow

On Edge1, the MRIB now shows an outgoing interface list that includes Border1.

```
Edge1#show ip mroute vrf Corp_VN 232.50.50.50 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.31.3, 232.50.50.50), 00:40:13/00:02:38, flags: sTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.195), Forward/Sparse, 00:40:13/stopped, Pkts:0,
p
      192.2.101.65, 00:40:13/00:02:38
```

In the underlay, not much is different:

```
Edge1#show ip mroute 232.0.3.195 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.70, 232.0.3.195), 00:42:40/00:03:08, flags: sTp
  Incoming interface: Null0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:42:40/00:03:08, Pkts:0, p
```


Aninda Chatterjee (aninchat@cisco.com)

Like before, Edge1 encapsulates the original multicast packet, with an outer destination IP address that matches the underlay SSM group that was derived for the overlay group.

Border1 is where the underlay multicast is slightly different:

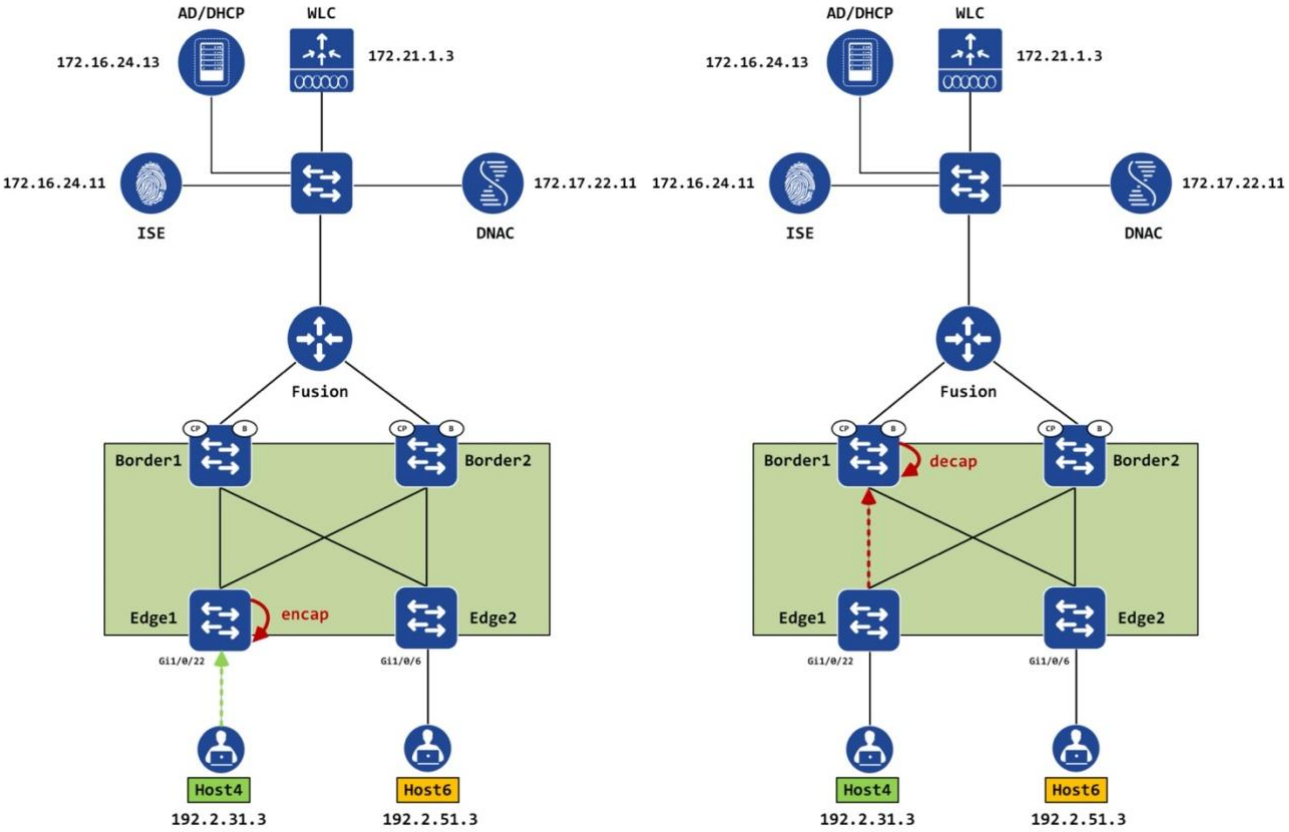
```
Border1#show ip mroute 232.0.3.195 verbose
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - 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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.70, 232.0.3.195), 00:45:04/00:02:55, flags: sT
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 192.2.101.69
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 00:45:04/stopped
```

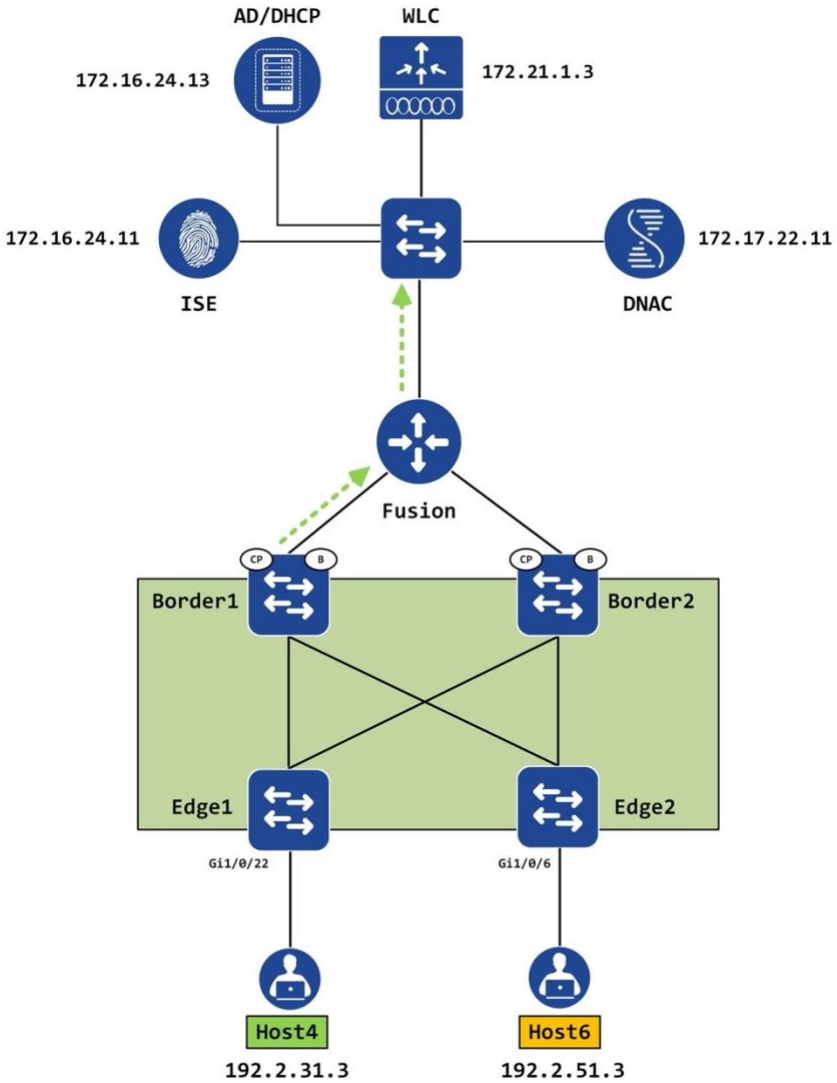
Because Border1 has a receiver attached off of it (technically, not directly, but it is in the path of the tree), it needs to decapsulate the packet. This is why the outgoing interface list points to Null0.

Once decapsulated, the packet is recirculated and another lookup is done in the VRF specific multicast table.

```
Border1#show ip mroute vrf Corp_VN 232.50.50.50 verbose
IP Multicast Routing Table

*snip*

(192.2.31.3, 232.50.50.50), 00:53:04/00:02:35, flags: sTp
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70,
232.0.3.195]
  Outgoing interface list:
    Vlan3014, Forward/Sparse, 00:53:04/00:02:35, p
```

This is now sent towards the Fusion and towards the receiver behind the shared services switch. The multicast data flow, thus, would be:



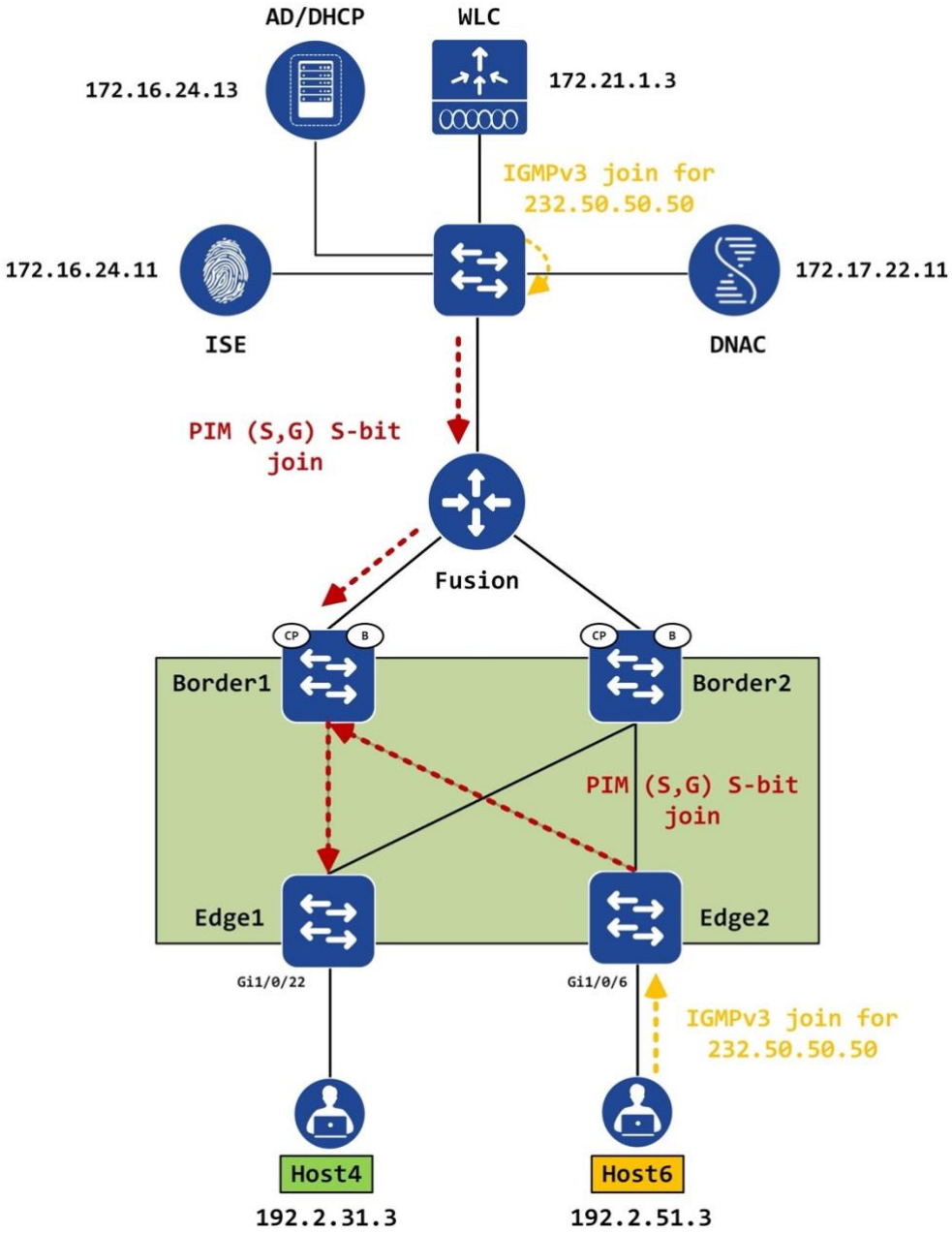


Aninda Chatterjee (aninchat@cisco.com)

SSM in SDA – source in fabric, receiver in and outside fabric

Receiver join and building SPT

Again, we'll quickly look at how the SPT is built. In this case, we have a receiver inside the fabric and a receiver outside the fabric. Both these receivers are configured to join 232.50.50.50, but from a specific source – 192.2.31.3. Last hop routers for both of these will initiate PIM (S,G) S-bit joins towards the source.



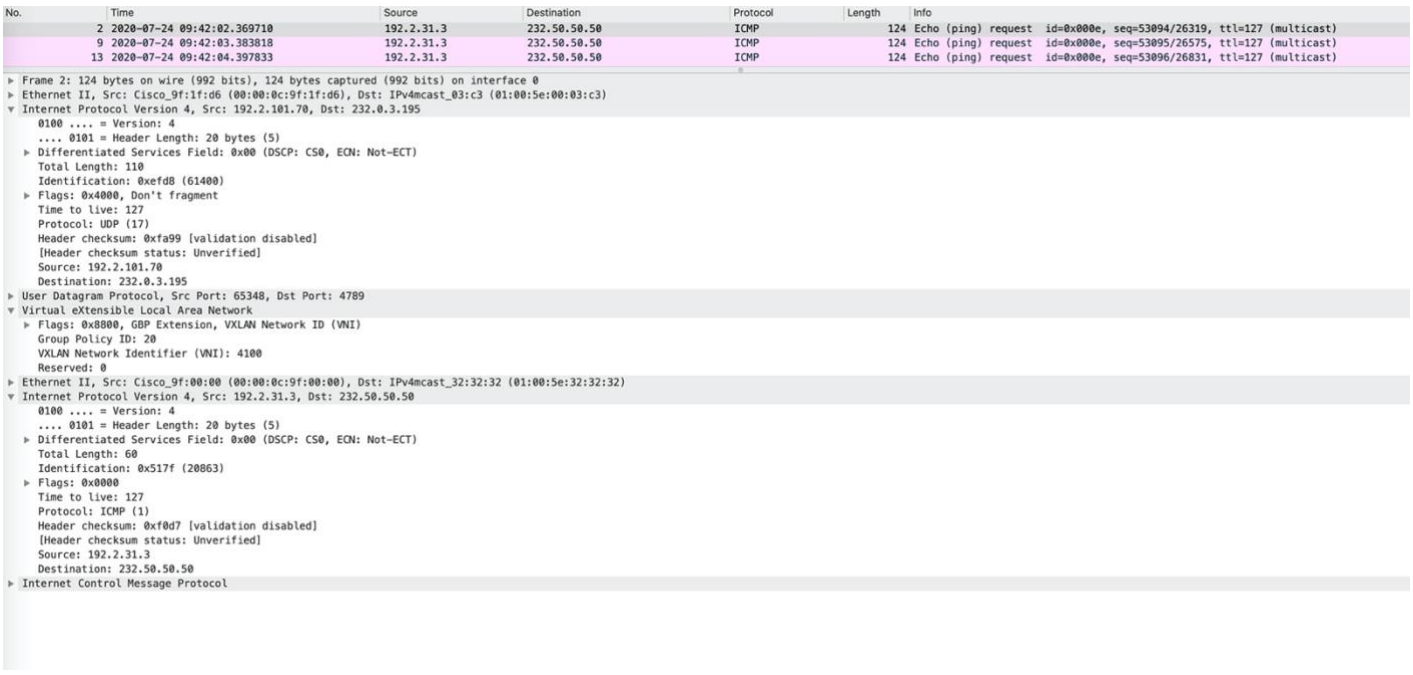
Aninda Chatterjee (aninchat@cisco.com)

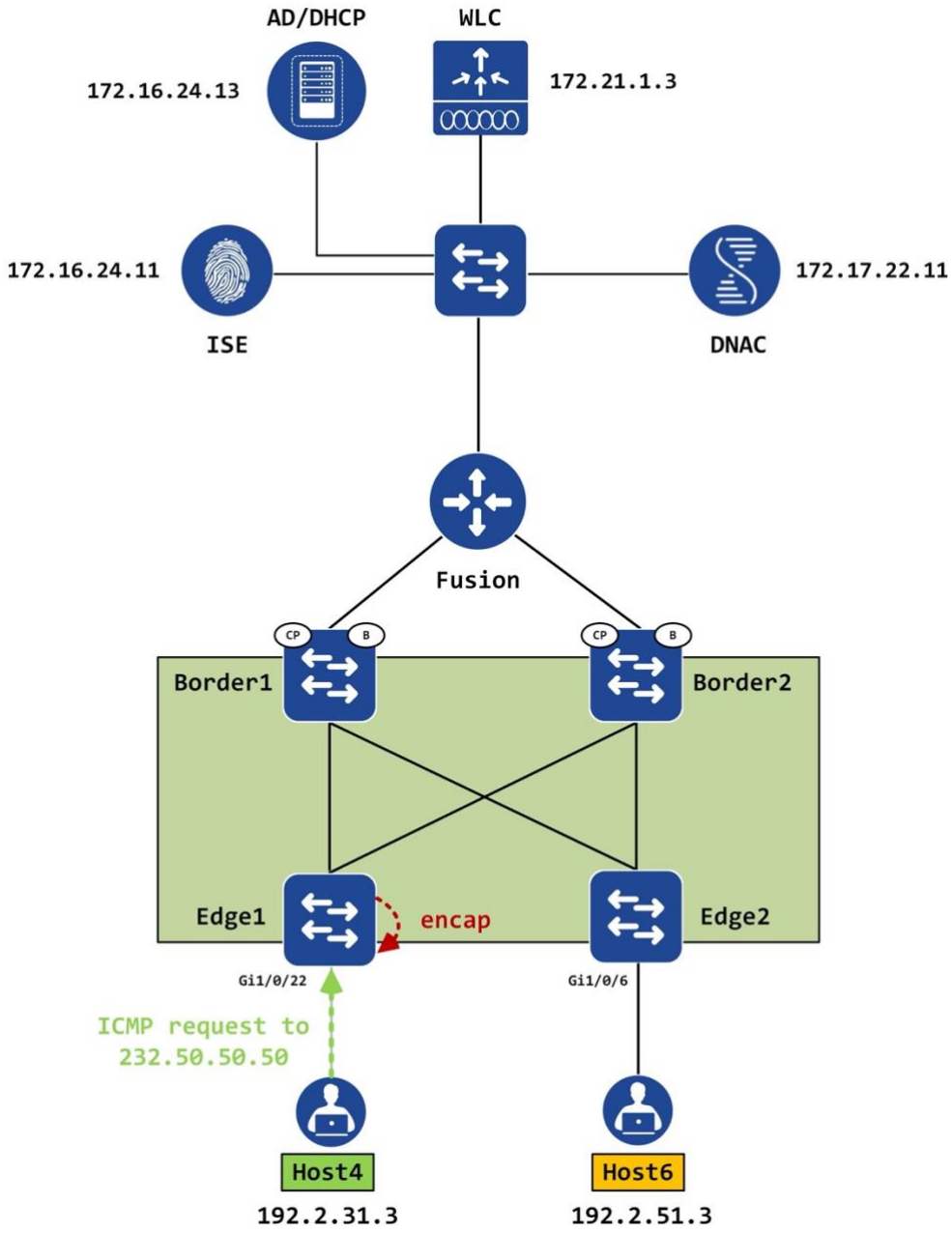
Final MRIB state and packet flow

There's not a lot that will be different in this case. Edge1 will continue to encapsulate the multicast data with the underlay SSM group that was derived for it.

```
Edge1#show ip mroute vrf Corp_VN 232.50.50.50 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.31.3, 232.50.50.50), 13:12:11/00:03:13, flags: sTp
  Incoming interface: Vlan1029, RPF nbr 0.0.0.0
  Outgoing interface list:
    LISP0.4100, (192.2.101.70, 232.0.3.195), Forward/Sparse, 13:12:11/stopped, Pkts:0,
p
      192.2.101.71, 10:51:37/00:03:13
      192.2.101.65, 13:12:11/00:02:31
```

The packet on the wire will be the same as what we've seen before:



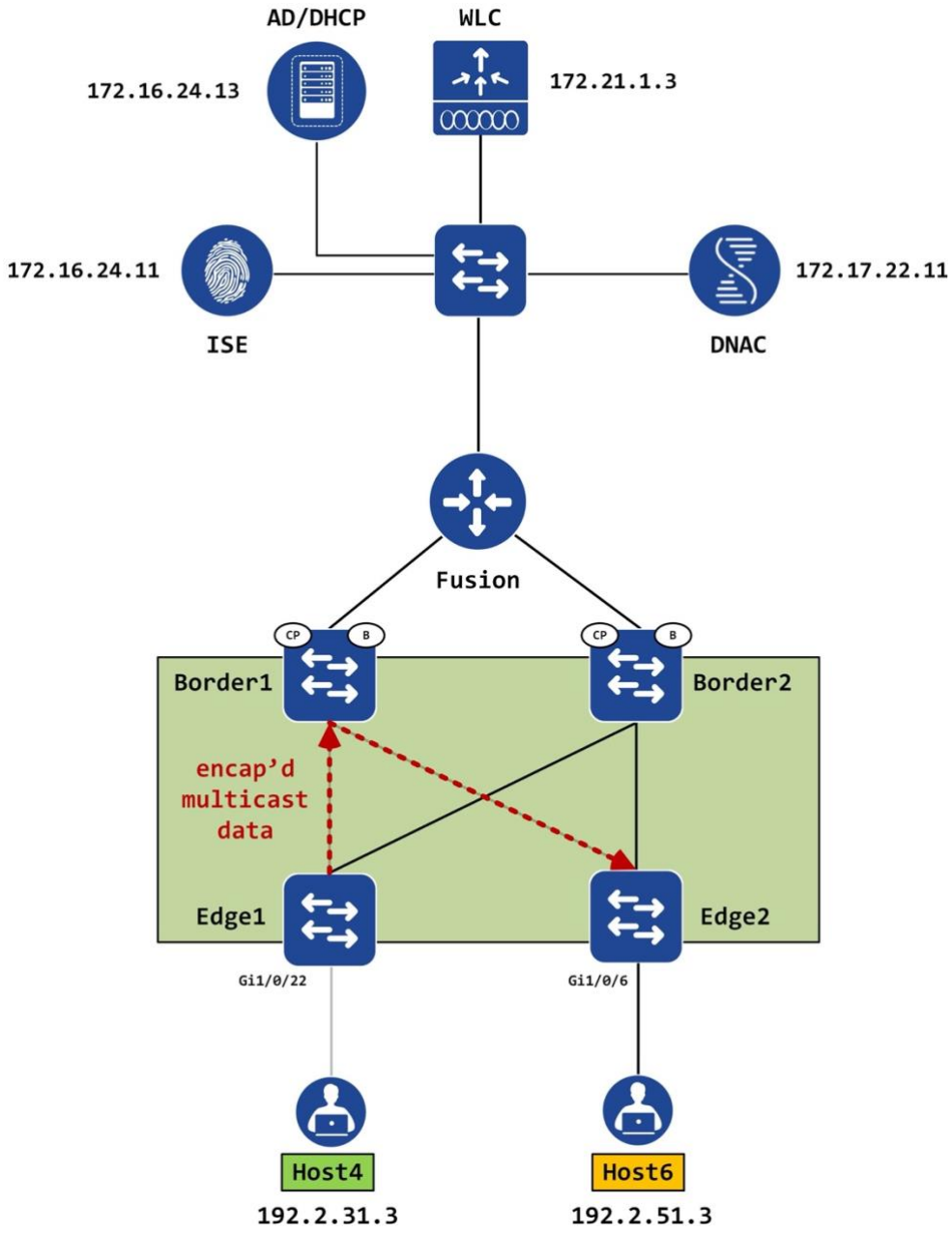


Border1 will have two entries in the outgoing interface list:

```
Border1#show ip mroute 232.0.3.195 verbose
IP Multicast Routing Table

*snip*

(192.2.101.70, 232.0.3.195), 14:02:51/00:03:00, flags: sTp
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 192.2.101.69
  lisp eid refcnt: 1
  Outgoing interface list:
    GigabitEthernet1/0/3, Forward/Sparse, 11:42:17/00:03:00, p
    Null0, Forward/Dense, 14:02:51/stopped
```

The outgoing interface tells us two things – Border1 will decapsulate the packet but it will also send a copy out of gig1/0/3.

Aninda Chatterjee (aninchat@cisco.com)



The decapsulated packet will be recirculated for another lookup, but this time in the VRF specific multicast table:

```
Border1#show ip mroute vrf Corp_VN 232.50.50.50 verbose
IP Multicast Routing Table

*snip*

(192.2.31.3, 232.50.50.50), 14:05:30/00:03:22, flags: sTp
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70,
232.0.3.195]
  Outgoing interface list:
    Vlan3014, Forward/Sparse, 14:05:30/00:03:22, p
```

This decapsulated packet will be sent towards the Fusion device and eventually towards the receiver behind the shared services switch.

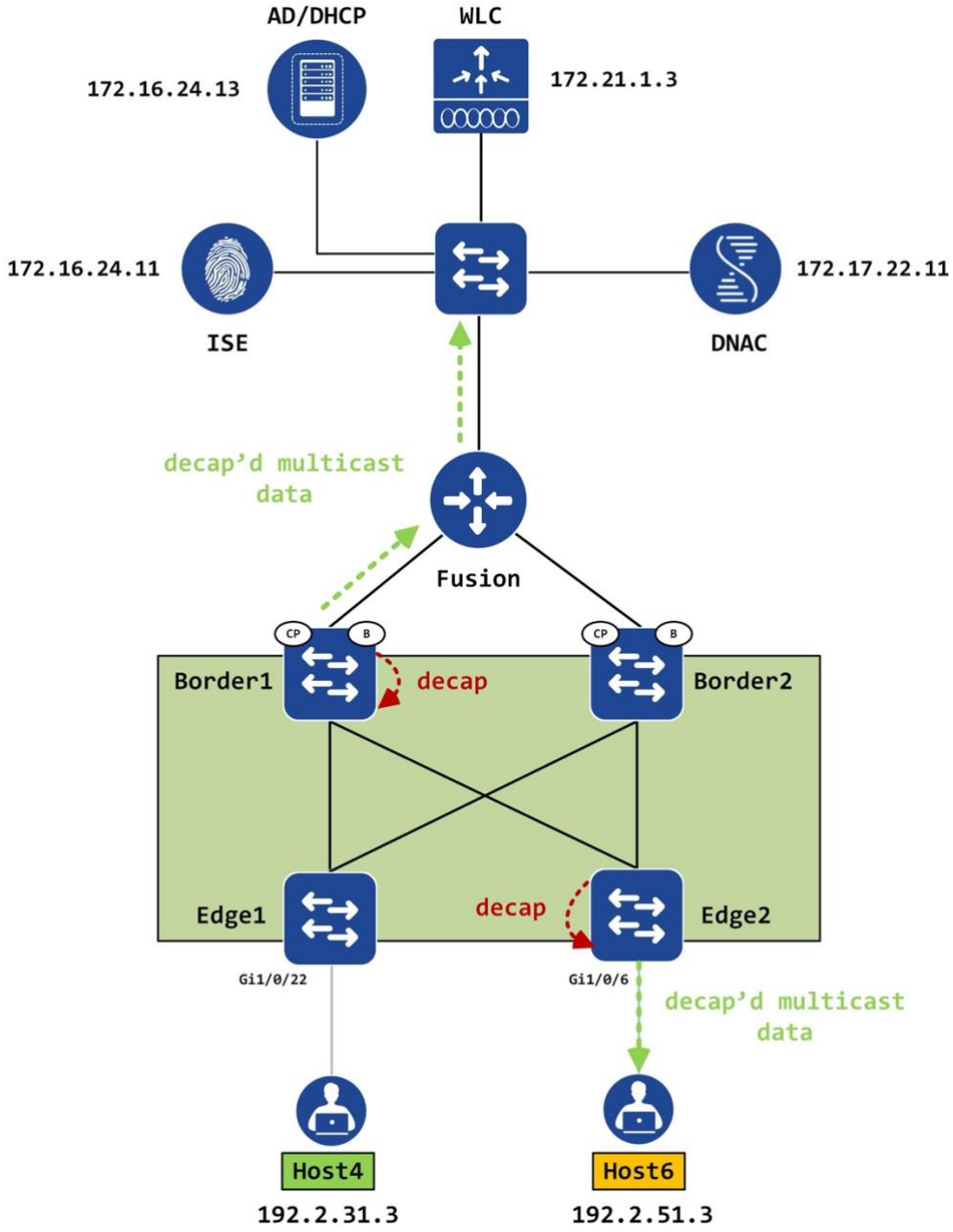
The packet that was replicated and sent to Edge2 will first be looked up in the underlay multicast routing table:

```
Edge2#show ip mroute 232.0.3.195 verbose
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
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(192.2.101.70, 232.0.3.195), 11:47:02/00:00:57, flags: sT
  Incoming interface: GigabitEthernet1/0/3, RPF nbr 192.2.101.72
  lisp eid refcnt: 1
  Outgoing interface list:
    Null0, Forward/Dense, 11:47:02/stopped, Pkts:0
```

The outgoing interface is Null0 which implies a decapsulation action. The decapsulated packet will be recirculated and lookup again, now in the VRF specific multicast table.

```
Edge2#show ip mroute vrf Corp_VN 232.50.50.50 verbose
IP Multicast Routing Table

*snip*

(192.2.31.3, 232.50.50.50), 11:48:26/00:02:56, flags: sTI
  Incoming interface: LISP0.4100, RPF nbr 192.2.101.70, LISP: [192.2.101.70,
232.0.3.195]
  Outgoing interface list:
    Vlan1030, Forward/Sparse-Dense, 11:48:26/00:02:56, Pkts:0
```


Aninda Chatterjee (aninchat@cisco.com)



Aninda Chatterjee (aninchat@cisco.com)