



# Cisco *live!*

January 29 - February 2, 2018 · Barcelona

BRKDCN-2099

# Multicast Deployment and Troubleshooting in Datacenter environment

Nagendra Kumar Nainar

Sr. Technical Leader

# Cisco Spark

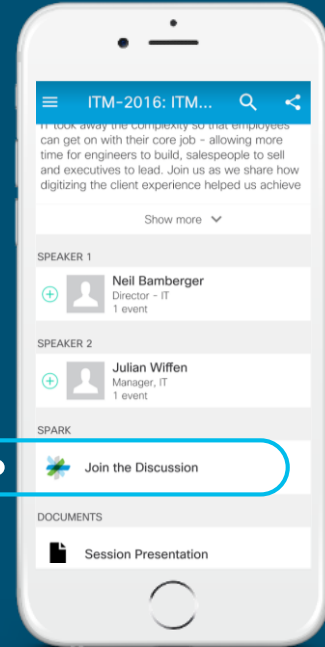


## Questions?

Use Cisco Spark to communicate with the speaker after the session

## How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space



[cs.co/ciscolivebot#BRKDCN-2009](https://cs.co/ciscolivebot#BRKDCN-2009)

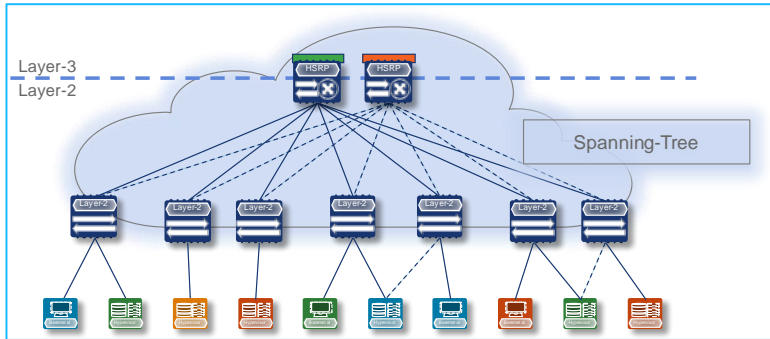
## Acknowledgement

- Matt Esau – Technical Leader, Cisco
- Richard Furr – Technical Leader, Cisco
- Alejandro Eguiarte – Technical Leader, Cisco

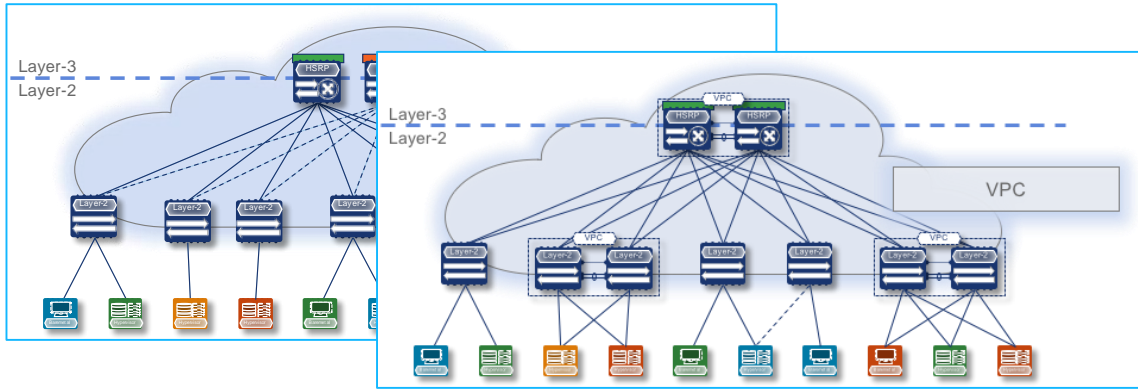
# Agenda

- Data center Network Evolution
  - Why Multicast in Data center
- NXOS Multicast Components
- NXOS Multicast Forwarding
  - State Creation and Packet Forwarding
- NXOS Multicast Troubleshooting
- Summary

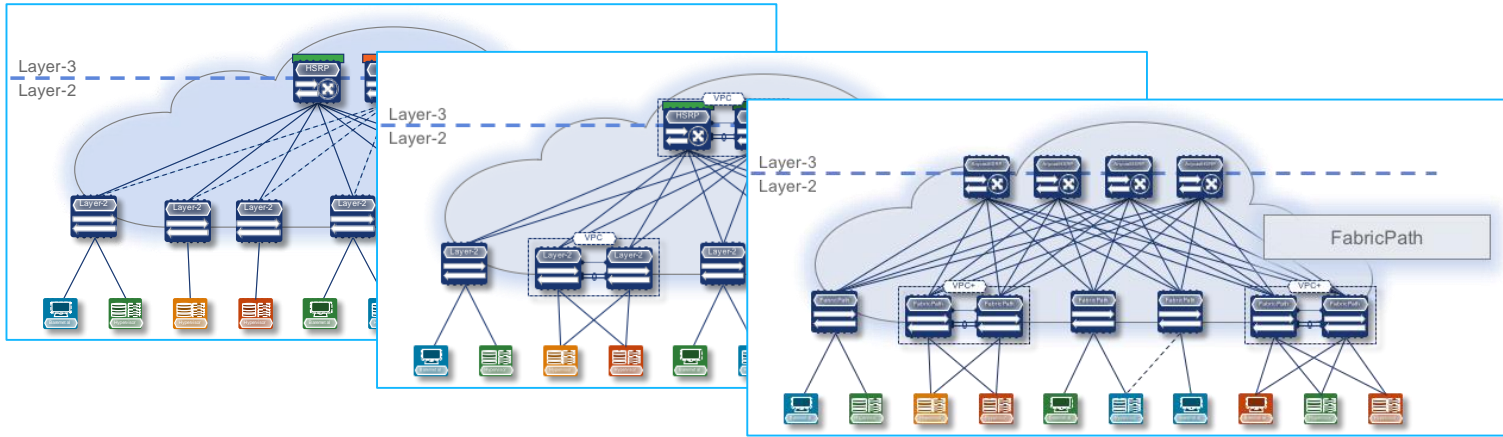
# Data Center Network Evolution



# Data Center Network Evolution

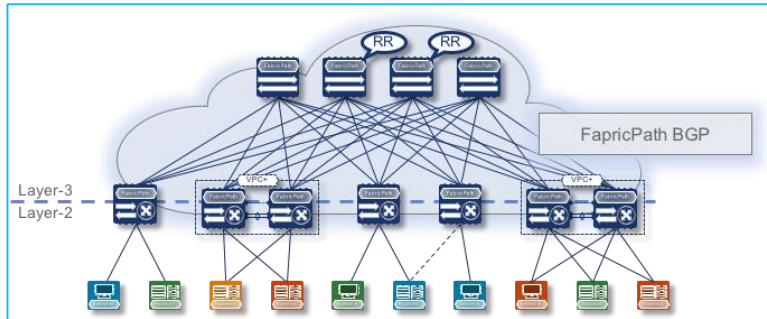
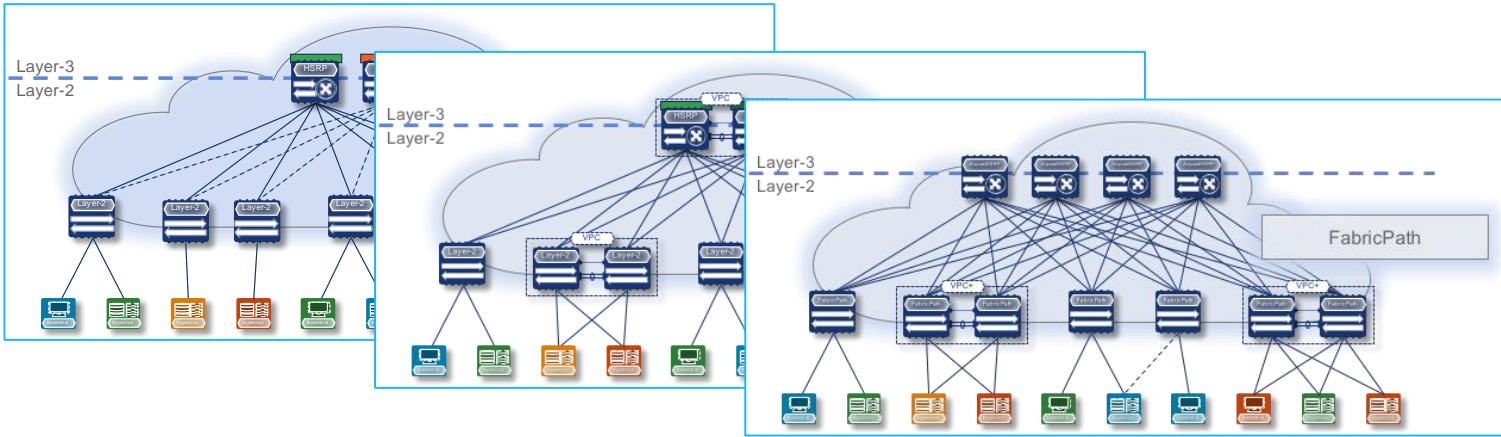


# Data Center Network Evolution

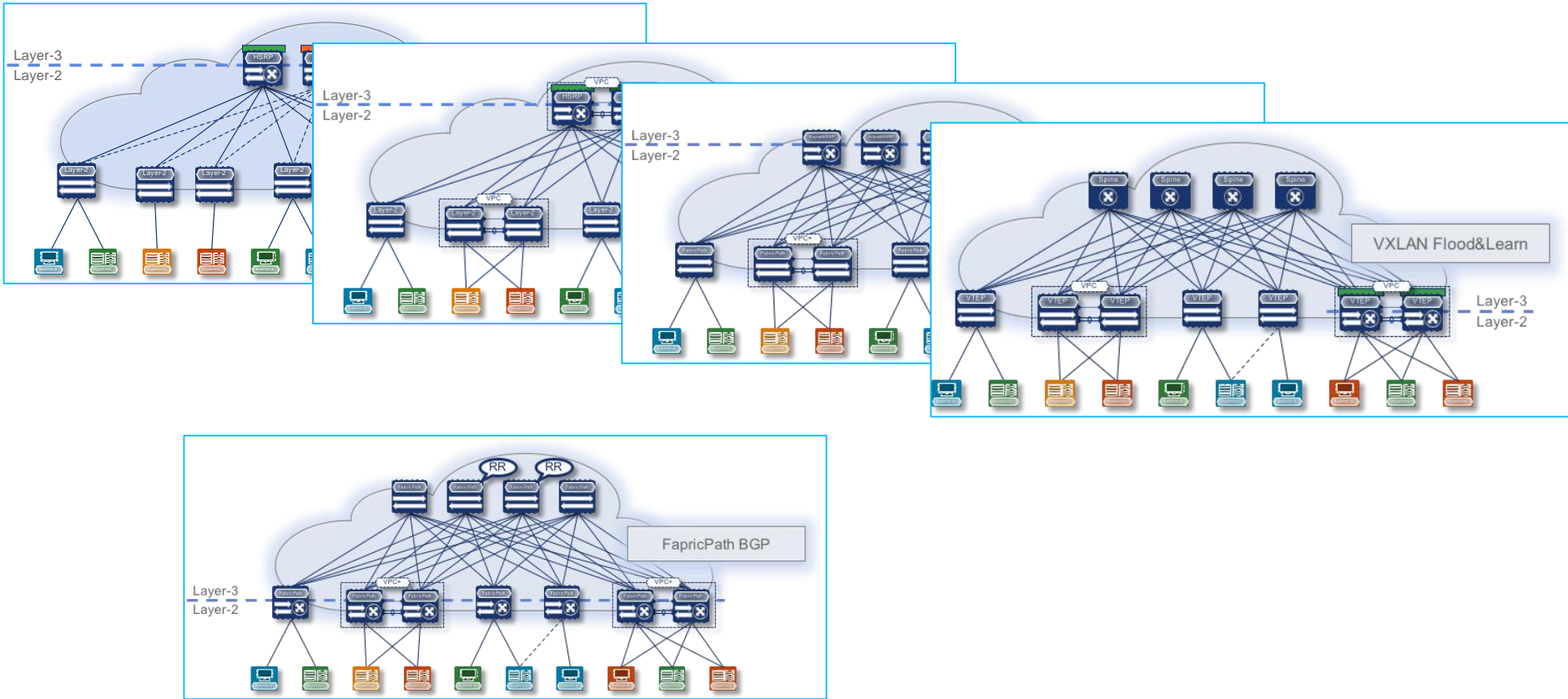




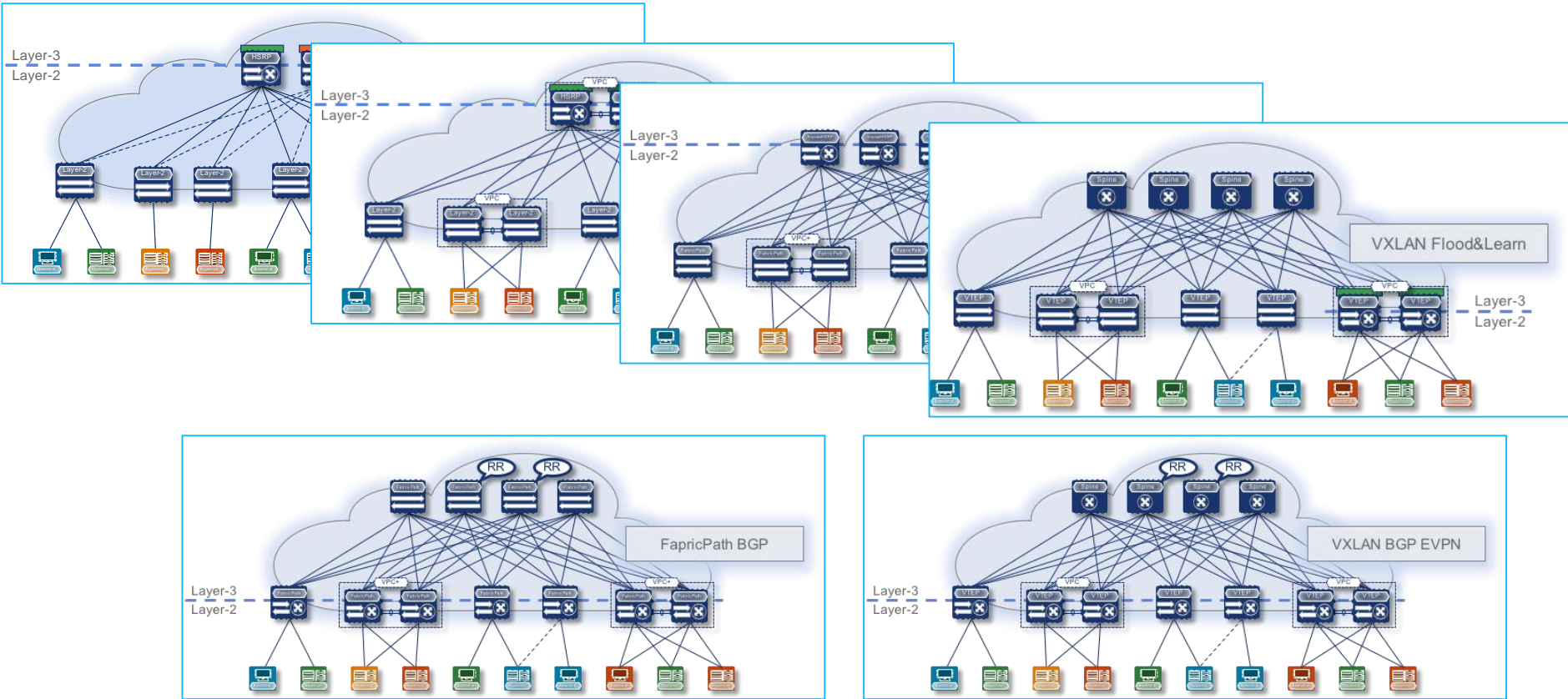
# Data Center Network Evolution



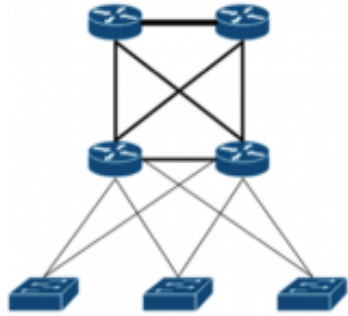
# Data Center Network Evolution



# Data Center Network Evolution



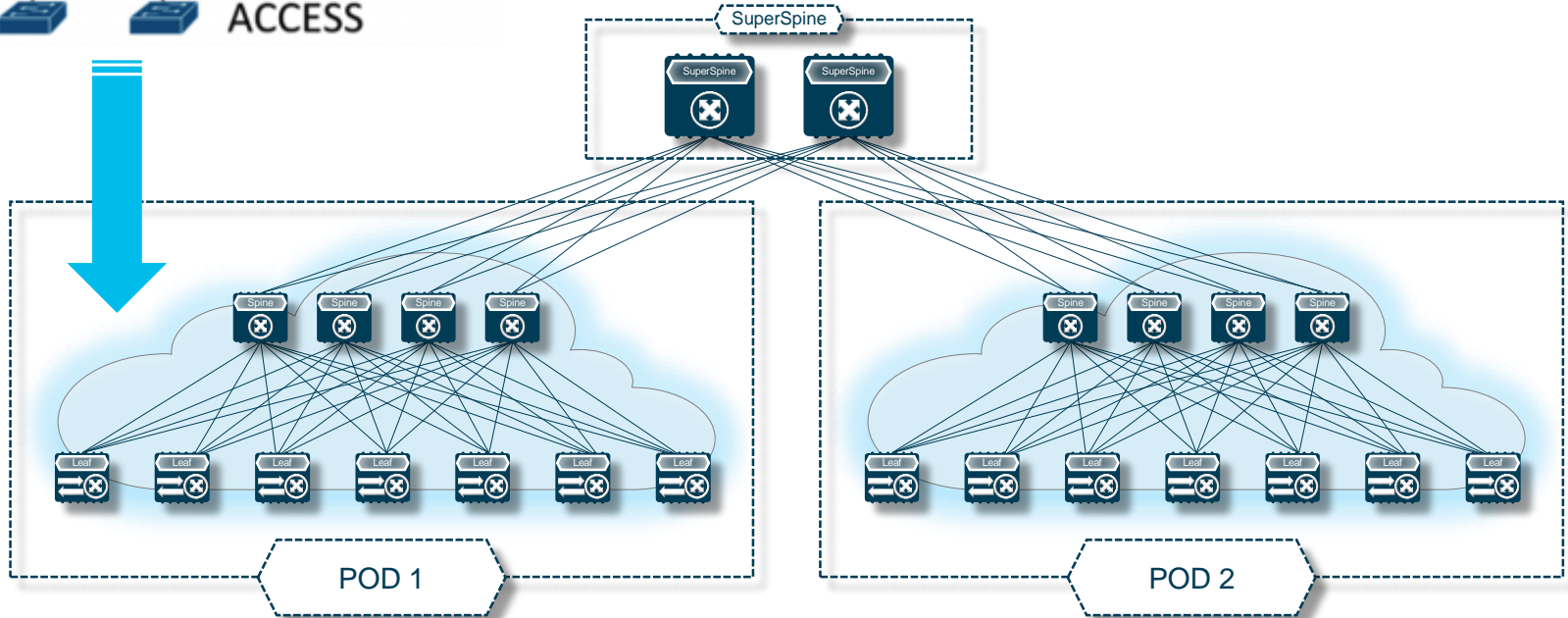
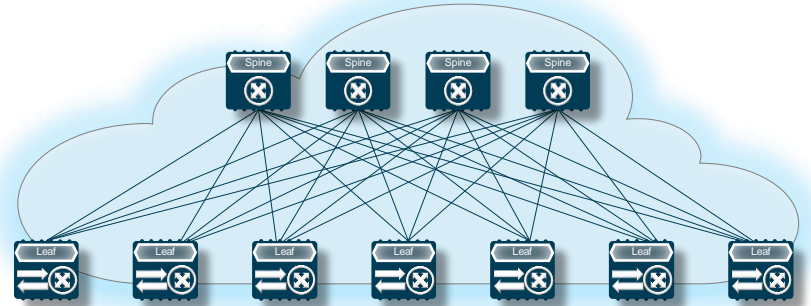
# Data Center Network Evolution



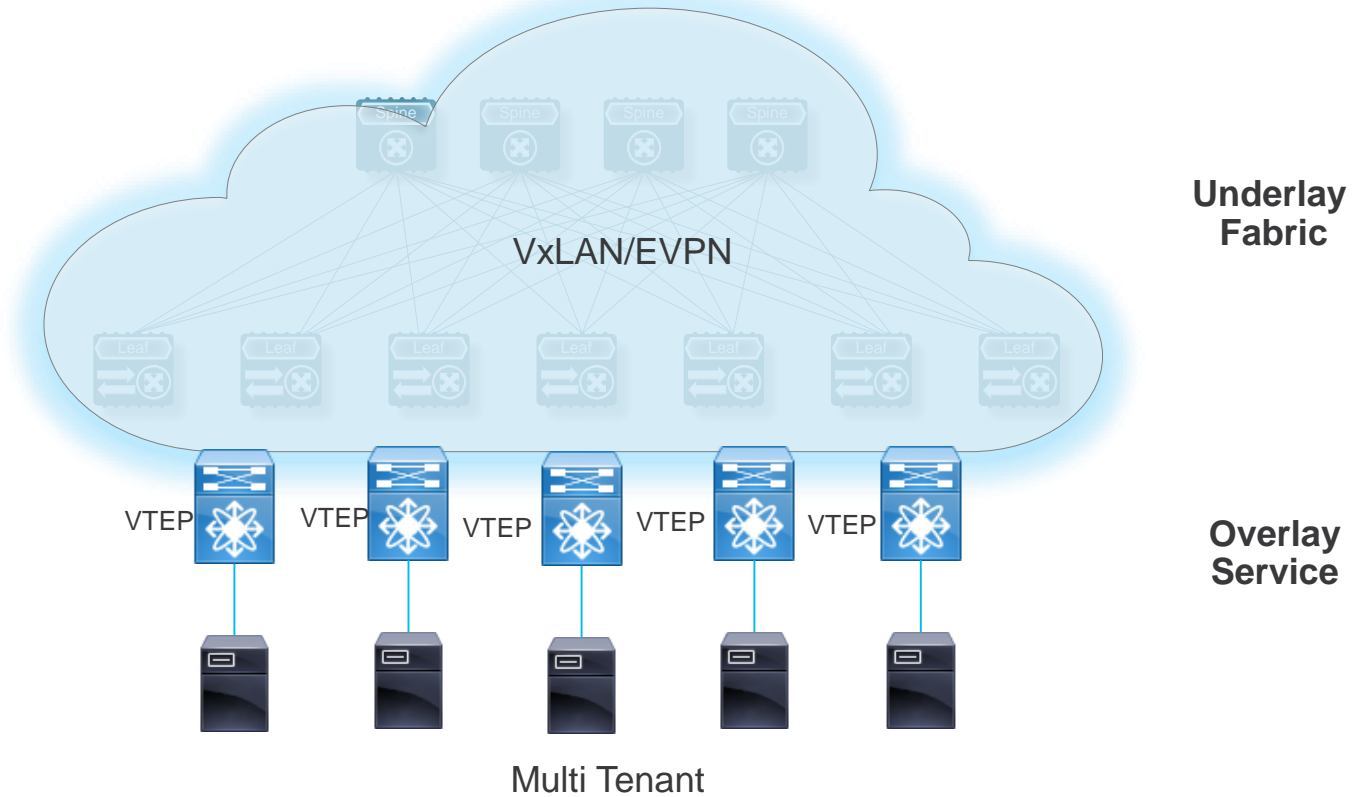
CORE

AGGREGATION

ACCESS

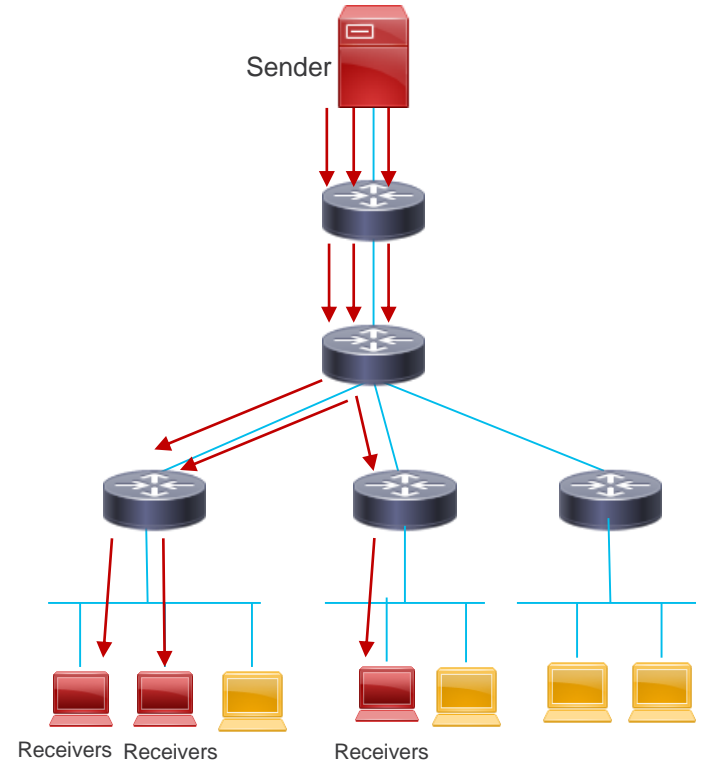


# Multi Tenant Data Center



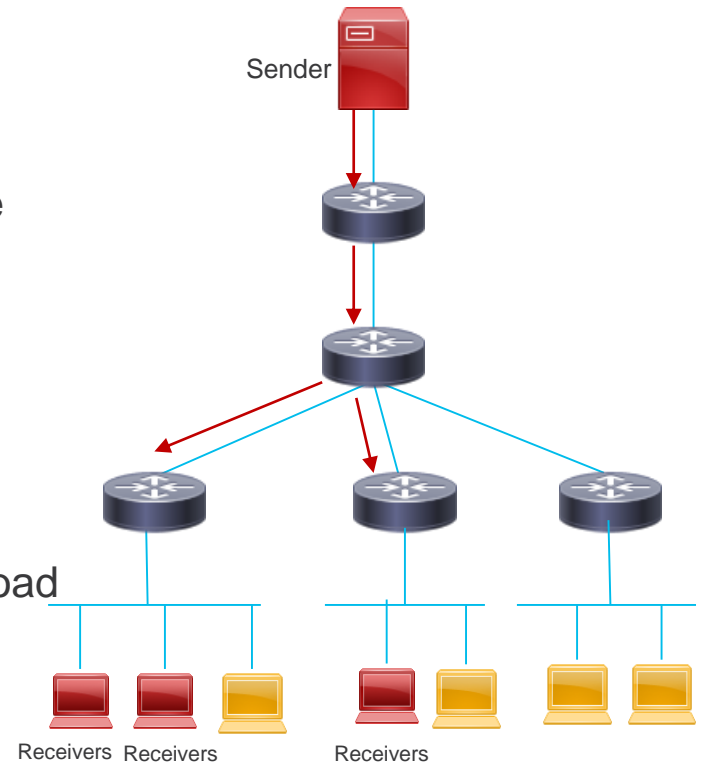
# Multicast Primer

- Inefficient End-Application performance
- Duplicate packet all over the network
- Inefficient Bandwidth Utilization
- Network Congestion
- Potential packet loss

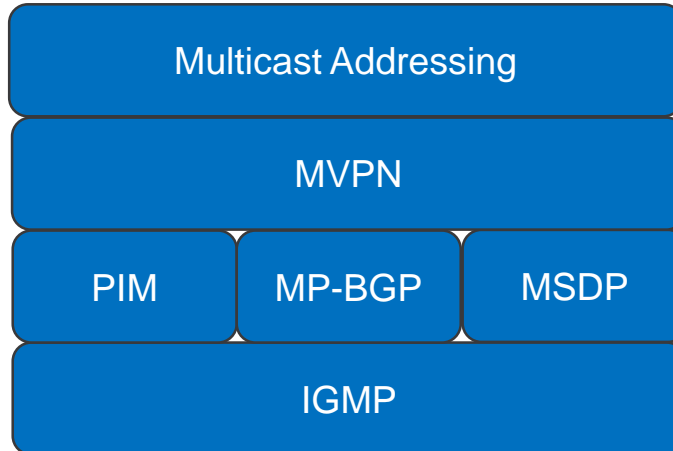
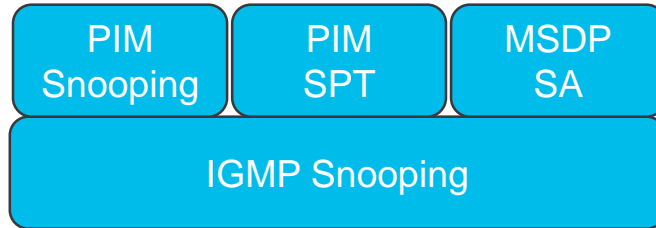


# Multicast Primer

- One-to-Many Data Communication model.
  - Same data stream sent from one source to multiple receivers
  - Packet replicated at the branching point.
- Efficient bandwidth utilization
  - Avoids duplication of data stream.
- Efficient End application Performance
  - Eliminates packet generation and processing overload
- Efficient Discovery mechanism
  - Dynamic protocol neighbor discovery



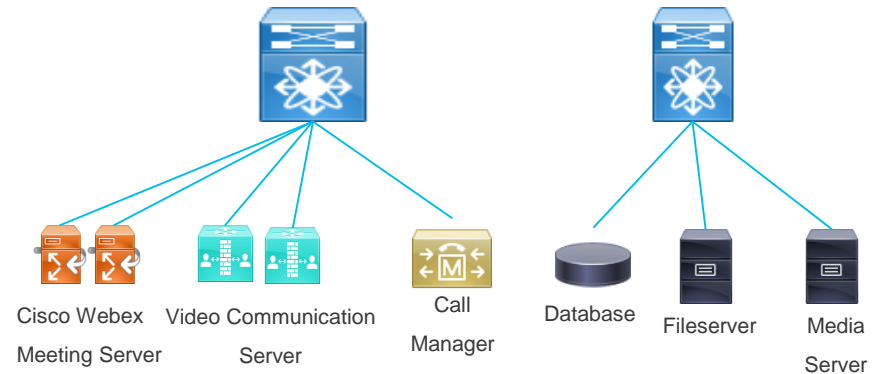
# Multicast Components





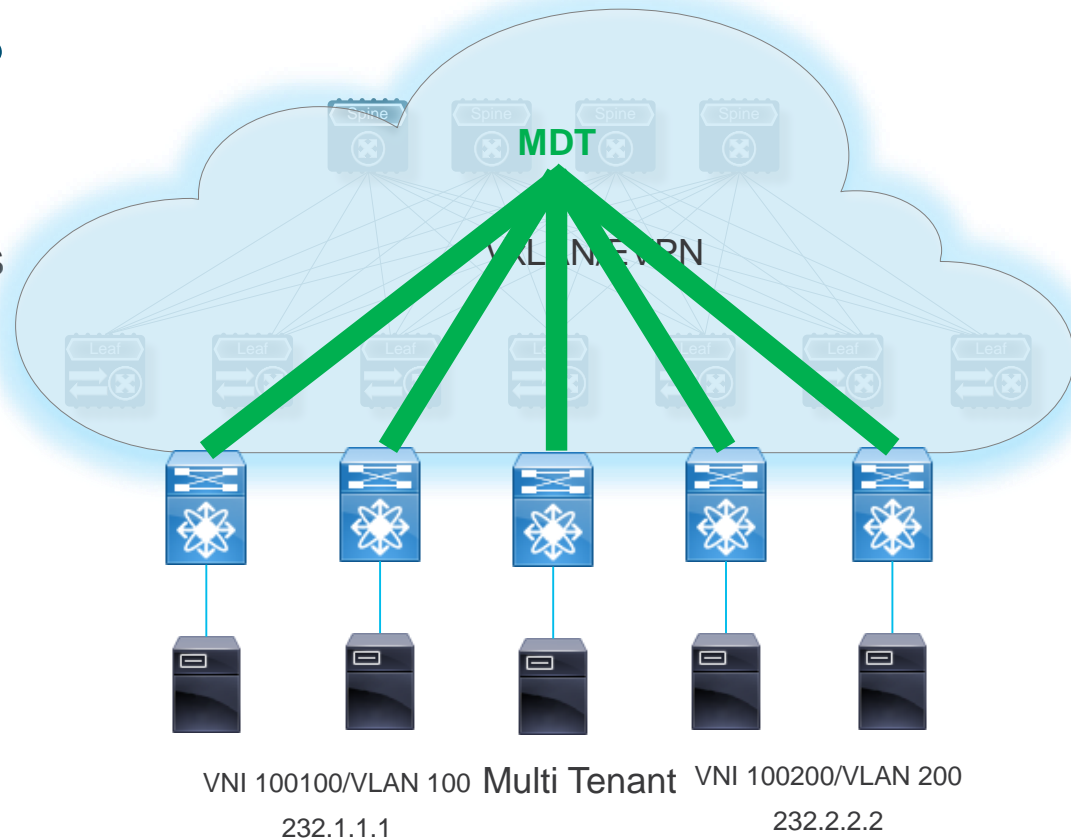
# Why Multicast in Datacenter?

- Various End Applications leverages multicast for data synchronization, backup etc.
  - Video and Collaboration Solutions
  - Distributed File systems
  - Data Replication and Synchronization
  - Media conferencing
  - Video Surveillance
- Common to see servers deployed in Datacenters.
  - Servers acting as multicast source
- Hosts can be senders or receivers.

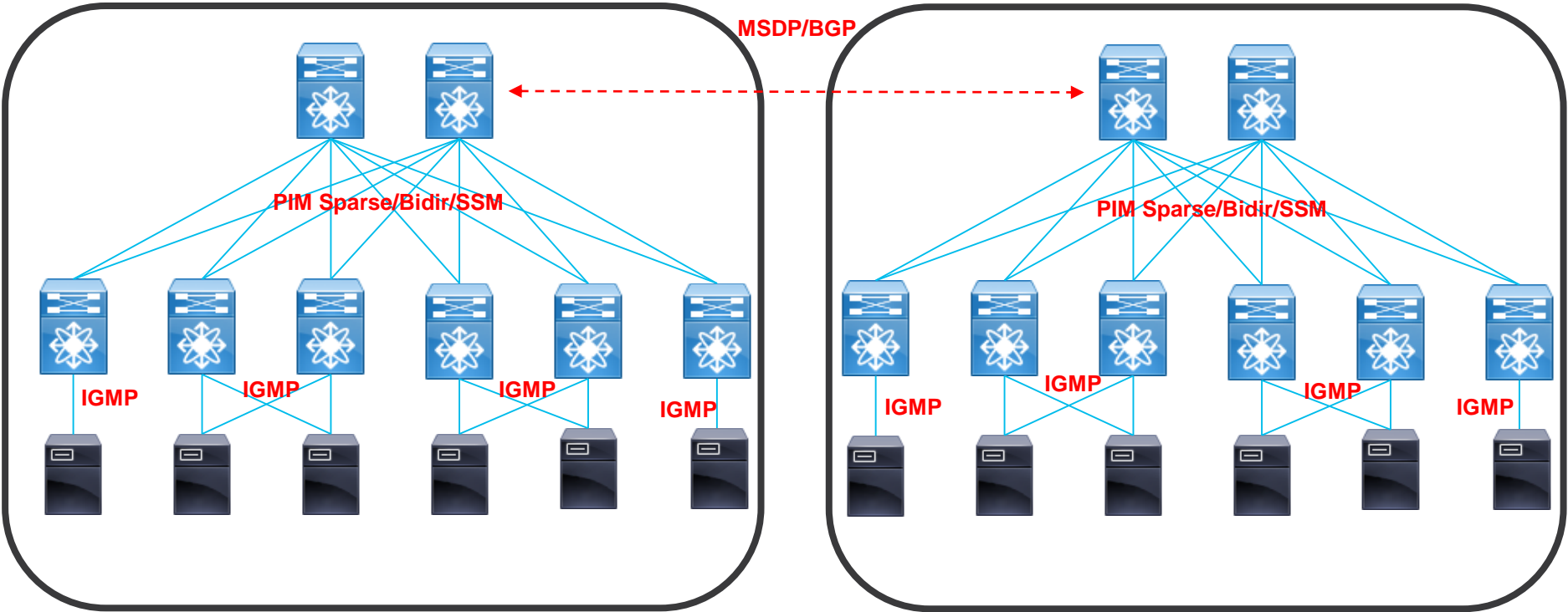


# Why Multicast in Datacenter?

- Provide multicast service to tenants
- Logical Layer 2 network spanning multiple sites.
  - ARP Query
  - Flood and Learn
- Ingress Replication vs Multicast
- Provides P2MP tunnel for Broadcast and unknown unicast traffic from tenants



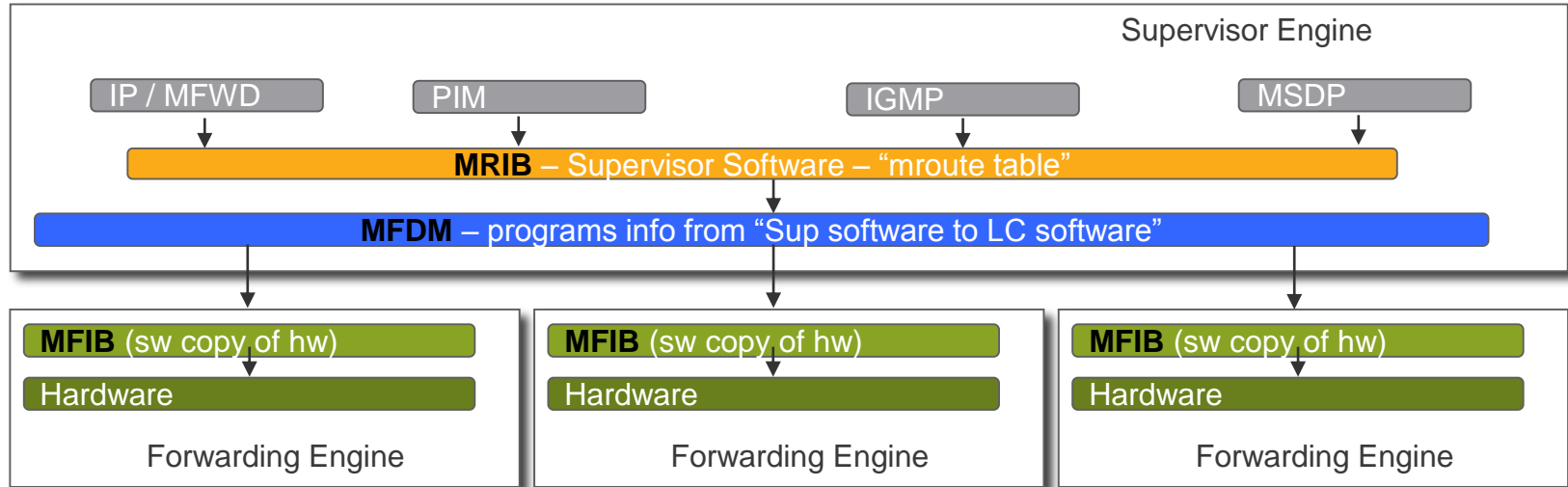
# Multicast Protocols in Datacenter



# NX-OS Multicast Architecture

- NX-OS is a modular operating system
  - Modularity helps with high availability, resource allocation and scale
- Some software components are always loaded
  - Others may be conditional
- Modularity includes multicast components
- Multicast in NX-OS is “VRF-Aware”
- NX-OS has unique features which can change traditional multicast models
- NX-OS does not support PIM Dense Mode

# NXOS Multicast components



# NXOS Multicast components

- **MRIB** (Multicast Routing Information Base – on by default)
  - Stores information from “client protocols” (PIM, IGMP, MSDP)
  - Provides detailed traffic statistics [Packets/Bytes]
  - Simply your Mroute Table from “`show ip mroute {detail}`”
- **MFIB** (Multicast Forwarding Information Base)
  - “Mroute table of the linecard” – pushes info down HW ASICs
  - Receives the copy of multicast entry from MFDM
  - **Problem Symptom:** “*My Mroute table looks correct, but not receiving packets*”

# NXOS Multicast components

- **MFWD** (Multicast Forwarding - on by default)
  - Responsible for SG creation
  - **Problem Symptom:** *“My SG entries aren’t being created on FHR.”*
- **MFDM** (Multicast Forwarding Distribution Module - on by default)
  - Route programming from MRIB to MFIB, and updates multicast statistic from MFIB -> MRIB
  - Acts as an interface between PI Sup process and Line card process.
  - **Problem Symptom:** *“My routes show (pending) status and traffic is not being forwarded”*

# NXOS Multicast components



- **M2RIB** (Layer 2 Multicast RIB)
  - Platform Independent process that handles Layer 2 Multicast forwarding details.
  - Derives OIF list for multi-destination frames
  - Programs the port state, incoming interface check, FTAGs etc
- **IPFIB**(on Nexus 7000)
  - Process runs on each I/O modules.
  - Responsible for (\*,G) and (S,G) programming in FIB/ADJ, OILS in MET
- **L2MCAST** (on Nexus 7000)
  - Process runs on each I/O modules.
  - Responsible for IGMP programming in MAC table



# Internet Group Messaging Protocol (IGMP)



```
NxOS# show ip igmp groups
IGMP Connected Group Membership - 2 total entries
Type: S - Static, D - Dynamic, L - Local, T - SSM Translated, H - Host Proxy
Group Address   Type Interface      Uptime  Expires  Last Reporter
233.1.1.1      D   Vlan200          00:13:47 00:03:09 192.168.200.6
233.1.1.1      D   Vlan100          00:10:08 00:03:19 192.168.100.23
NxOS#
```

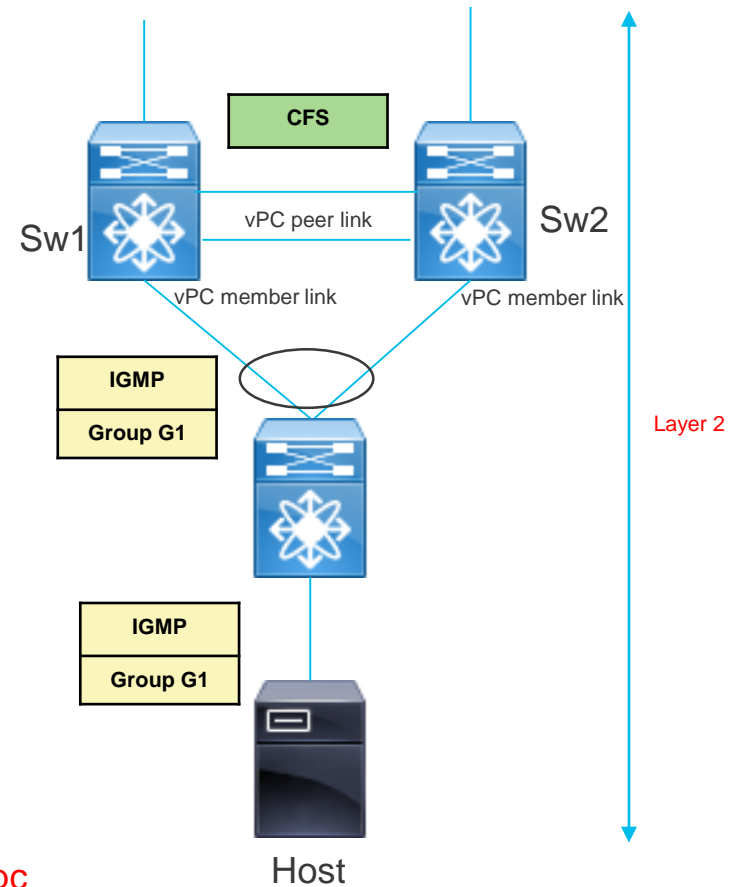
- Host-Router group membership protocol to signal the interest to receive a multicast stream
- IGMP Messages are as below:
  - Query
  - Membership Report
  - Leave
- IGMP functionality on DC switches are as below:
  - Send Queries periodically on all PIM enabled interfaces.
  - Process join/leaves and program the forwarding table accordingly.
  - Keep track of all host (on a per group basis) that are sending membership reports.

# IGMP Snooping

- IGMP Snooping is enabled by default on Nexus platforms
- 224.0.0.X are link-local multicast address and reserved for protocol use.
  - All switches should **flood** the frame with destination IP address 224.0.0.X
- All frames with destination MAC **0100.5E00.00XX** will be flooded. Avoid using IP multicast groups that map to this MAC address range
- Detect mrouter ports via IGMP query and PIM hello
- Can be configured as IGMP V3 querier. Support hosts running all IGMP version with backward compatibility
- Fast leave is disabled by default
- IGMP v3 explicit tracking is on. Track joins from individual host

# IGMP in vPC environment

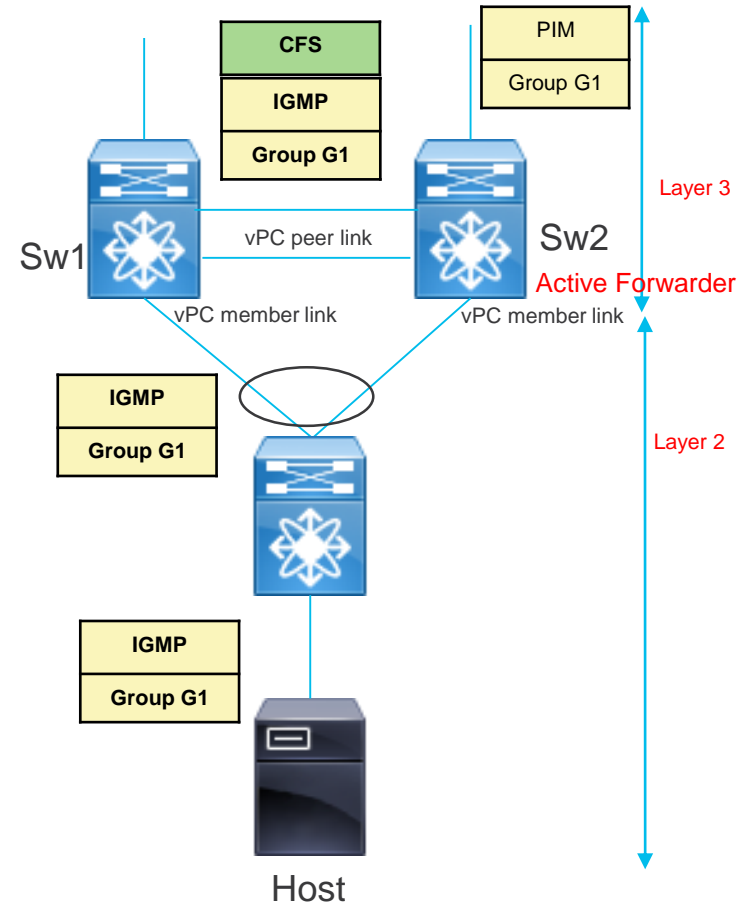
- When vPC peers are Layer2, IGMP snooping table will be in sync
- IGMP message that is snooped by one vPC peer will be synced with other peer.
- IGMP Join/Leave messages received on vPC member ports will be synced with peer.
- vPC peer uses CFS messages to sync the state.
- IGMP snooping must be enabled



Show ip igmp snooping internal event-history vpc

# IGMP in vPC environment

- When vPC peers are Layer3, IGMP state entries will be in sync.
- IGMP message that is received by one vPC peer will be forwarded to other peer with CFS encapsulation.
- One of the vPC will become active forwarder and send PIM message towards source/RP
- Active forwarder is chosen based on the best metric.
- CFS is used to negotiate the role.



# IGMP Snooping in VxLAN environment

- VxLAN Gateway will flood all control and data traffic over all ports.
- Primarily applicable for Nexus 9000 platforms
- Supported from 7.0(3)I5(1) release.
- IGMP snooping over VxLAN is not supported for FEX ports
- ARP-ETHER TCAM is required to be configured as double wide.

```
feature nve
ip igmp snooping vxlan
hardware access-list tcam region arp-ether 256 double wide
```

# Static IGMP

- Some deployment can benefit with Static IGMP
  - End Application does not support IGMP
  - End Application is always a receiver
  - End application Clusters
- LHR can be configured to be a receiver or hardware switch to OIL.

```
interface vlan 100
ip igmp join-group <> source <>
```

—————> LHR will be a receiver

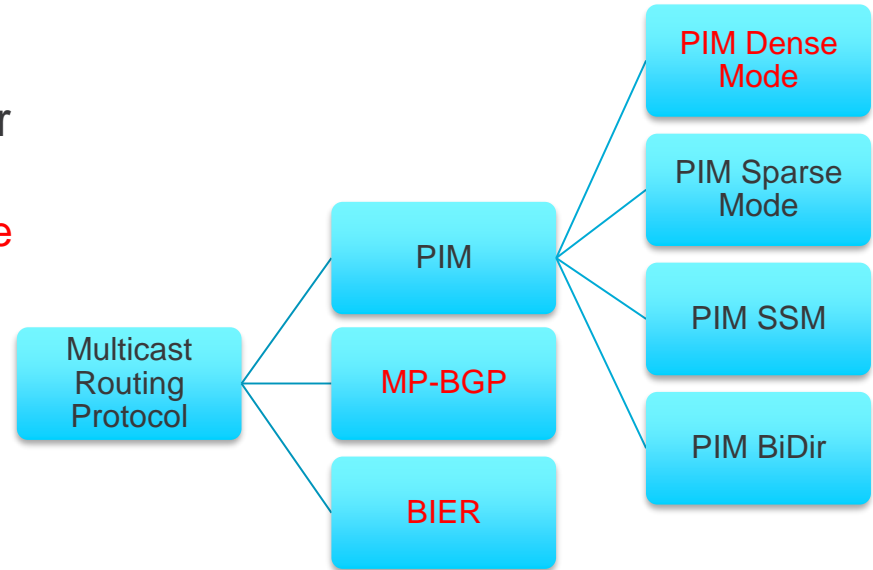
```
interface vlan 100
ip igmp static-oif <> source <>
```

—————> Hardware forwarding

- Static IGMP requires static IGMP snooping for the respective VLAN

# Multicast Routing Protocol

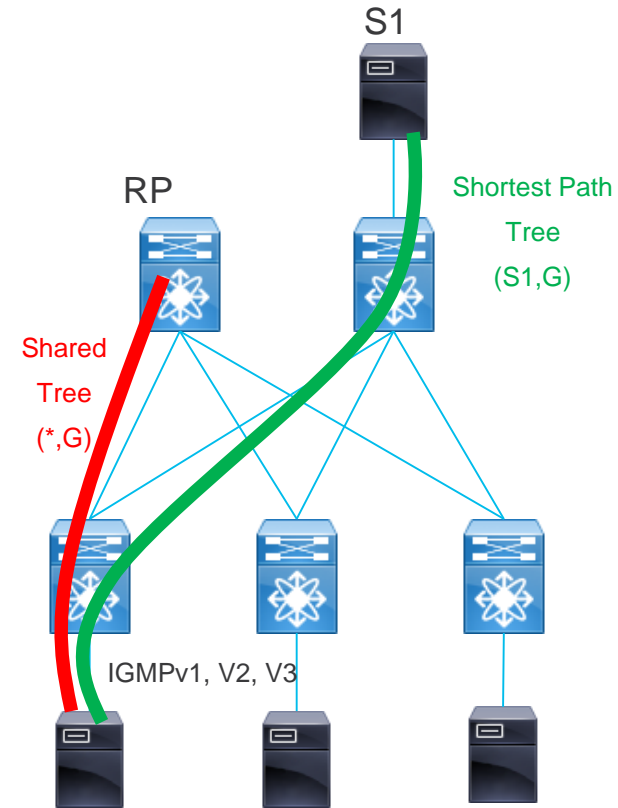
- PIM is the commonly deployed multicast routing protocol in Datacenter.
- PIM dense mode is not a viable option for Datacenter
  - Nexus switches does not support **PIM dense mode**.
- MP-BGP is a scalable Overlay signaling protocol for multicast.
- BIER is a new stateless multicast architecture.
- MP-BGP and BIER is not currently supported in Datacenter switches



# PIM Any Source Multicast (ASM) Mode

- MDT is built explicitly from receivers towards RP/Source.
- Uses shared tree and optionally uses source-based trees
- $(*,G)$  entries are created based on control plane.  $(S,G)$  entries are created based on data traffic.
- FHR uses PIM Register message to register the source to RP.
- Can support arbitrary source and receiver distribution
- Group membership tracked via IGMPv1, v2, or v3

```
ip pim use-shared-tree-only
interface vlan 100
 ip pim sparse-mode
```

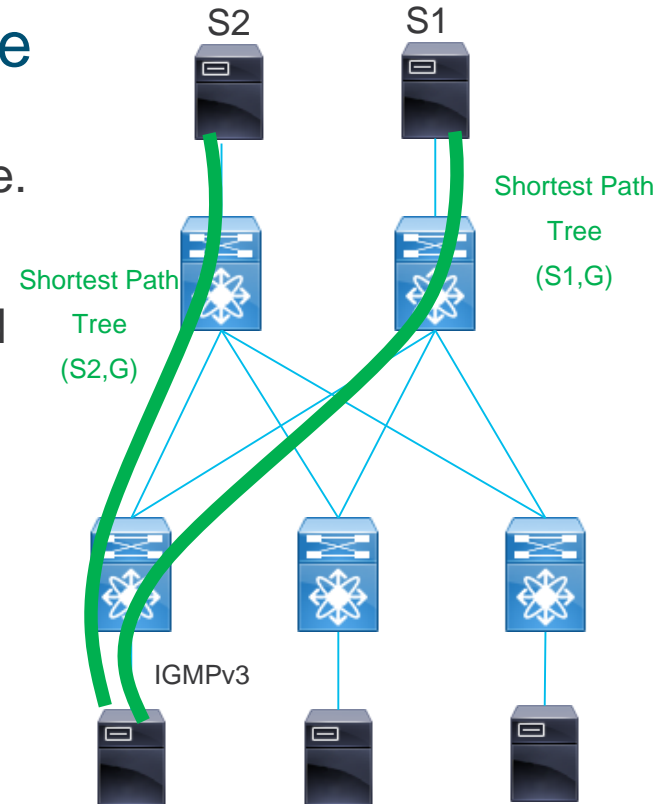




# PIM Source-Specific Multicast (SSM) Mode

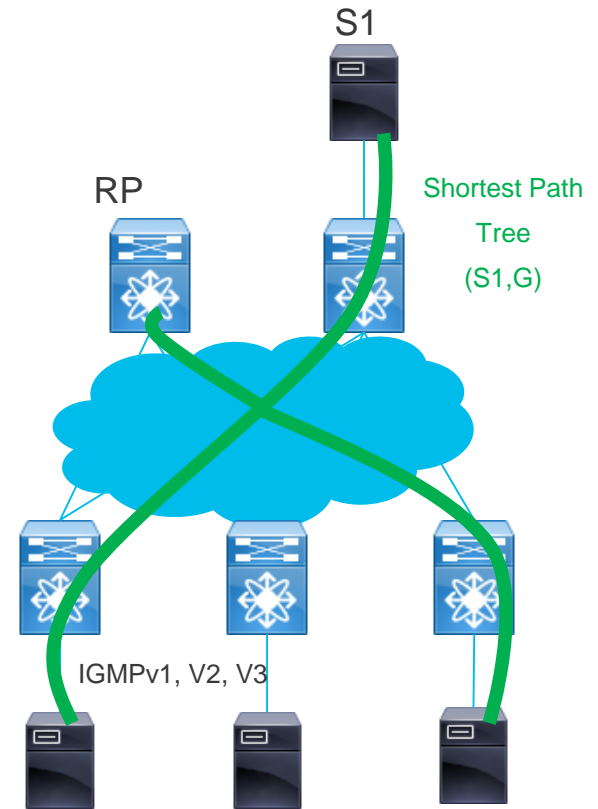
- MDT is built explicitly from receivers towards Source.
- Uses source-based trees.
- Group membership should request both Source and Group.
- No (\*,G) entries. (S,G) entries are created based on control plane.
- Uses OOB mechanism for Source discovery
- Group membership tracked via IGMPv3.

```
ip pim ssm range 233.0.0.0/8
interface vlan 100
 ip pim sparse-mode
```



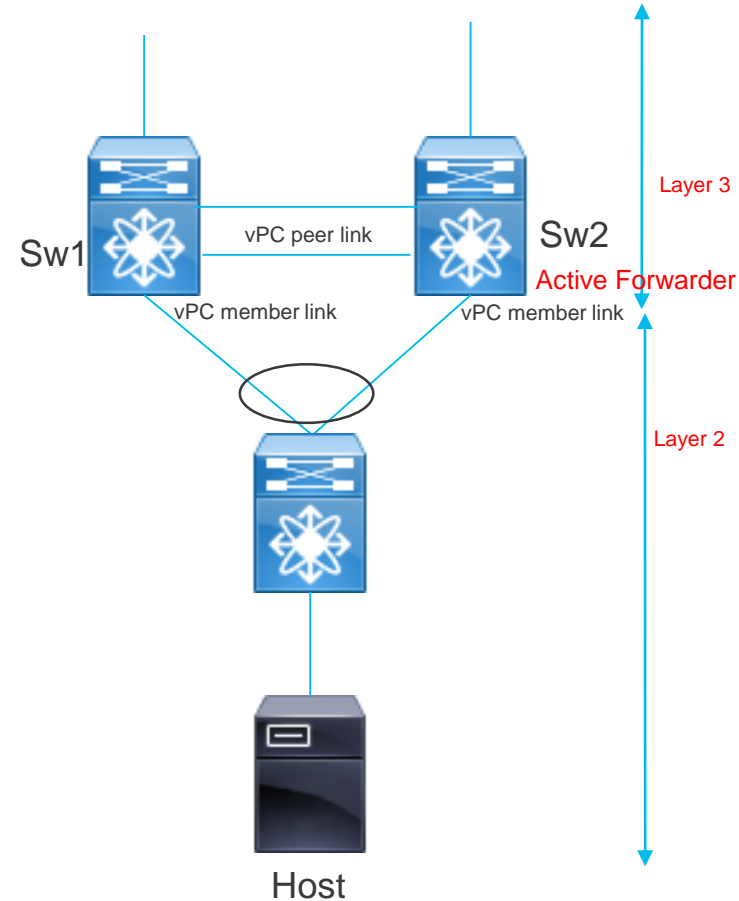
# PIM Bidir Mode

- Multipoint-to-Multipoint MDT is built towards RP.
- Massively scalable—ideal for many-to-many applications
- Data-flow independent—no registers, asserts, non-RPF issues
- Drastically reduces network mroute state
  - Eliminates ALL (S,G) state in the network for Bidir groups
  - Shortest path trees from sources to RP eliminated
  - Source traffic flows both up and down shared RP tree
  - Permits virtually unlimited sources



# PIM in vPC environment

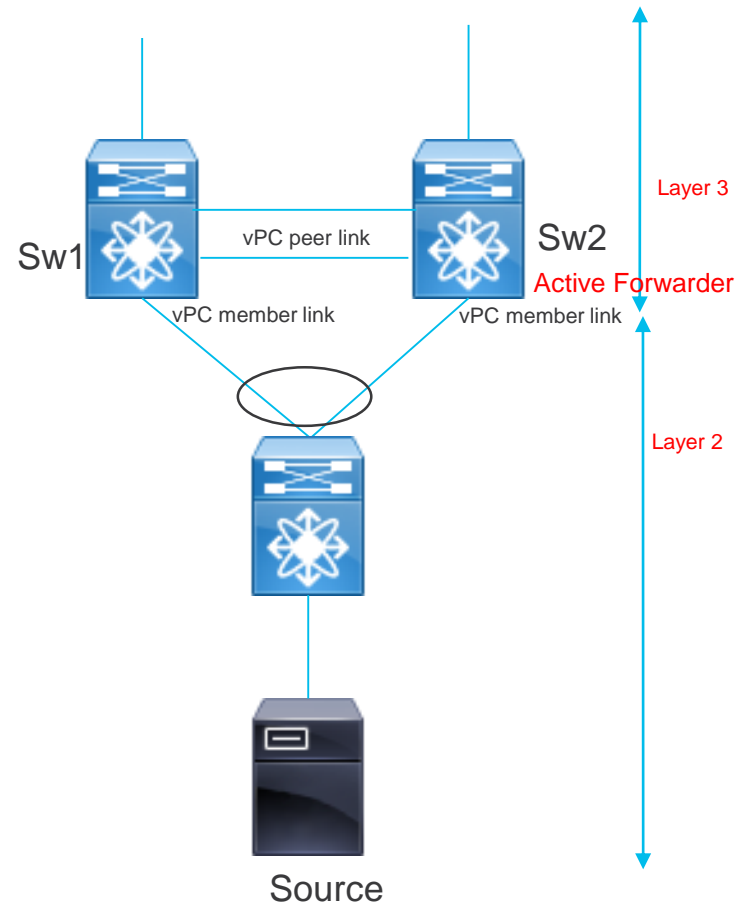
- PIM ASM Mode is the only supported mode on Nexus platforms.
  - Nexus 9000 supports ASM and SSM
- vPC peers use CFS messages to elect the Active Forwarder.
  - vPC peer close to RP/Source will be elected as Active Forwarder.
- PIM neighborship can be established between vPC peers.
- PIM neighborship cannot be established over vPC member links



# PIM in vPC environment

## Register Behavior

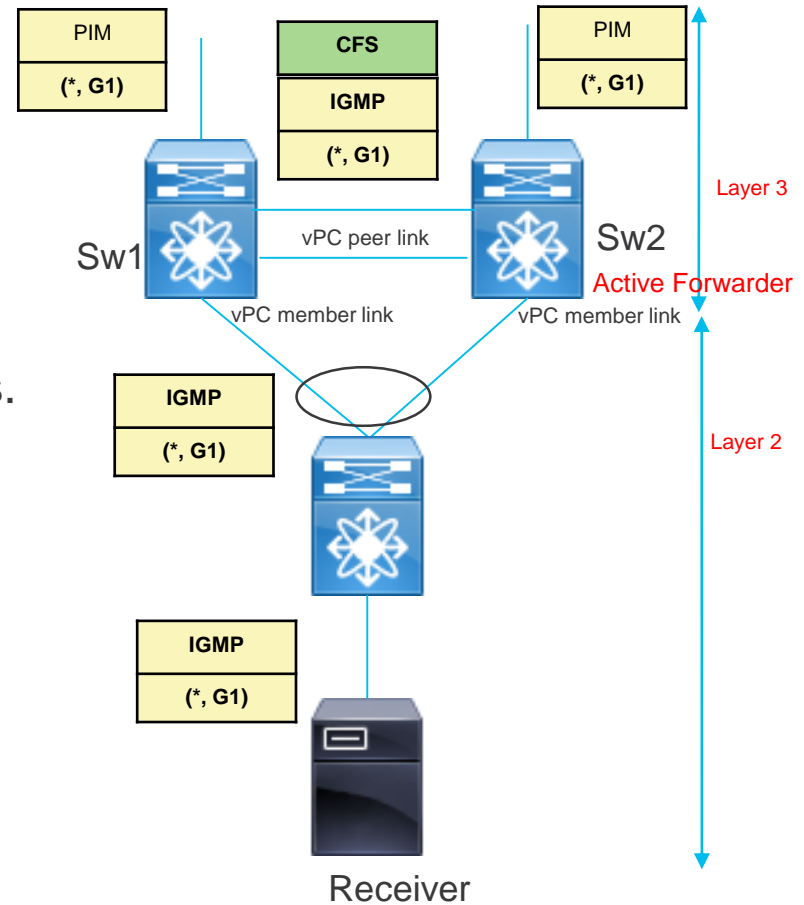
- First Hop Router connecting the source is vPC peer.
- Also known as Ingress vPC
- One of the vPC peer will be the PIM DR
- vPC peer will forward the data traffic from source over peer-link.
- vPC peers will create (S,G) state entry.
- Only PIM DR will register the source with RP.



# PIM in vPC environment

## Join Behavior

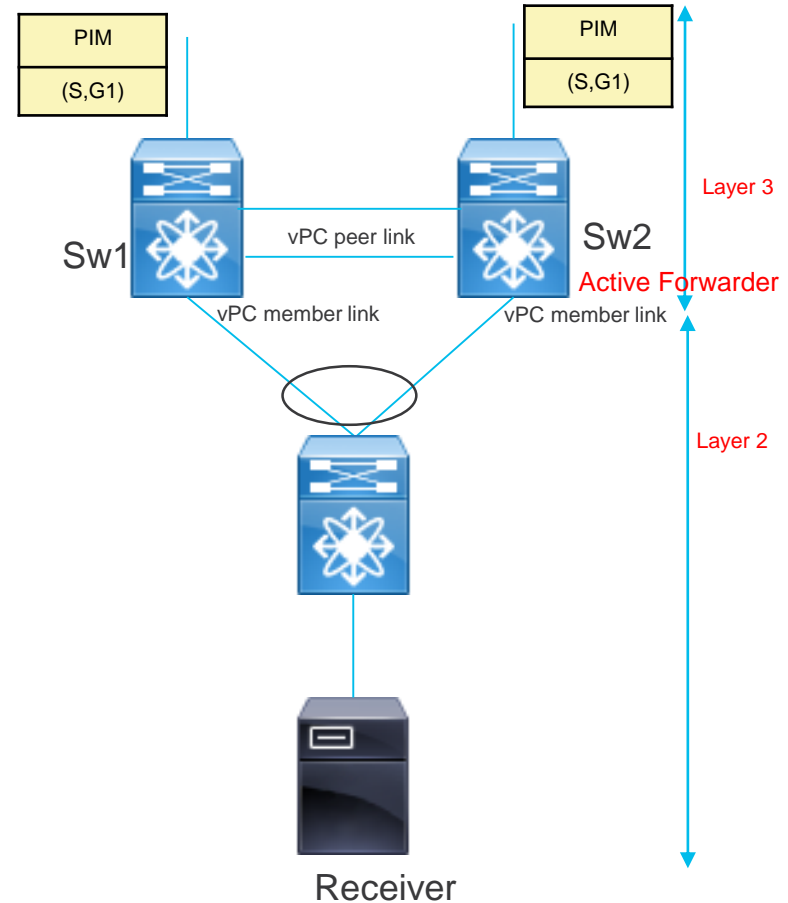
- Last Hop Router connecting the receiver is vPC peer.
- (\*,G) entry will be created on both vPC peers.
- Both vPC peer builds \*,G towards RP.
- Both vPC peers builds S,G towards Source.
- vPC peers uses CFS to elect the Active Forwarder.



# PIM in vPC environment

## Pre-built SPT

- By default, only the Active forwarder will build SPT by sending PIM join.
  - Convergence delays upon forwarder-change
  - S,G expiry on non-forwarder may cause duplication
- Pre-build SPT on non-forwarder by triggering upstream PIM J/Ps (without OIFs)
- Traffic pulled always, and dropped due to no OIFs
- Feature not enabled by default in vPC
  - A cli-knob “ip pim pre-build-spt” is required.
  - Per vrf context



# PIM in vPC environment

## Pre-built SPT

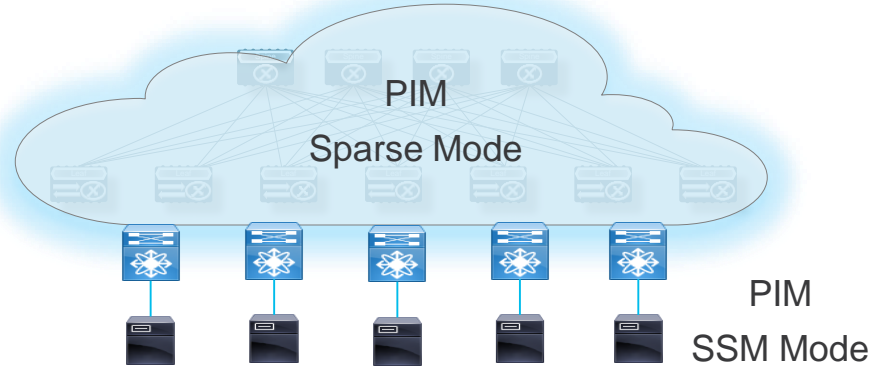
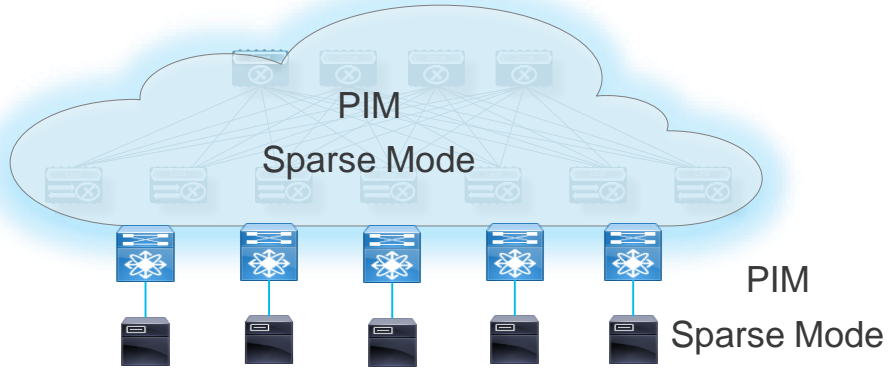
### Benefits

- Avoids temporary duplicates due to periodic expiry of (S,G) routes on the non-forwarder, causing forwarding on (\*,G)s, until a (S,G) is created again. The pre-build SPT on the non-forwarder ensures (S,G) routes are built and maintained even when not forwarding traffic.
- Pre-builds the SPT multicast tree on the non-forwarder router and removes the convergence delay needed to restore traffic when a forwarder role change is triggered
- Not relying on PIM for restoring traffic in DR/ RPF Interface failure cases. As Best Practice default PIM Hello Timers can be used with no convergence impact in these failure scenarios.

### Things to be aware

- Consumption of bandwidth on parallel links between the VPC pair & the upstream source.
- Upstream multicast routers should have ability to do 2x the multicast replication as without pre-building the SPT.

# PIM underlay/overlay



- Underlay and Overlay Multicast are independent to each other.
- Different PIM modes can be used for Overlay and Underlay
- Currently SSM is not supported for VxLAN
- Border nodes should be enabled with both underlay and Overlay multicast



# Underlay Multicast Configuration

## PIM

```
feature ospf
feature pim
ip pim rp-address 10.1.4.4 group-list 232.0.0.0/8
```

```
interface Ethernet1/4
no switchport
ip address 10.1.111.1/24
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode
no shutdown
```

```
NxOS1# show ip pim neighbor
PIM Neighbor Status for VRF "default"
Neighbor          Interface          Uptime    Expires    DR          Bidir-   BFD
                  Interface          Uptime    Expires    Priority    Capable  State
192.168.100.2     Vlan100           00:01:48  00:01:21  1          yes      n/a
NxOS1#
```

# Overlay Multicast Configuration

## PIM

```
feature nv overlay
nv overlay evpn
feature pim
```

```
vrf context TEST
  ip pim rp-address 10.1.4.4

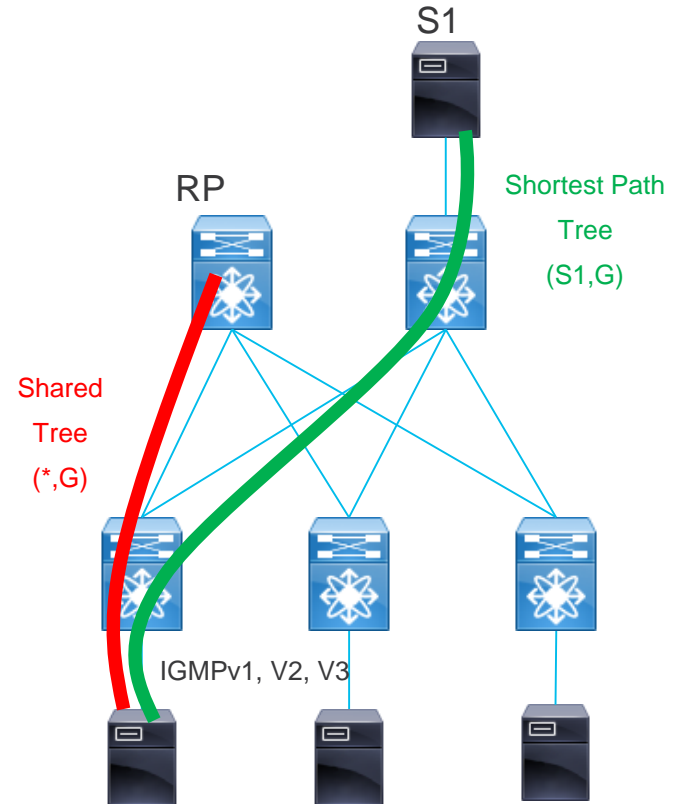
interface vlan 100
no shutdown
vrf member TEST
ip address 192.168.100.1/24
ip pim sparse-mode
```

```
interface nve1
no shut
host-reachability protocol bgp
source-interface loopback 0
member vni 100100 mcast-group
239.1.1.1
```

```
NxOS1# show ip pim neighbor vrf TEST
PIM Neighbor Status for VRF "TEST"
Neighbor          Interface          Uptime    Expires    DR          Bidir-   BFD
                  Interface          Uptime    Expires    Priority    Capable  State
192.168.100.2    Vlan100           00:01:48  00:01:21  1           yes      n/a
NxOS1#
```

# PIM Rendezvous Point (RP)

- RP acts as the shared root for multicast shared tree.
- RP is not required for PIM SSM mode.
- LHR builds  $(*,G)$  towards RP.
- FHR registers the source for any stream with RP.
- Single point of failure. Needs redundancy consideration.



# PIM Rendezvous Point (RP)

- Multicast group can be scoped as ranges
- Various RP deployment options available:
  - Static RP
  - Auto-RP
  - BSR
- Redundancy is a primary consideration for PIM RP.
  - Anycast RP
- vPC peer can act as a PIM RP if the PIM state entries can be synchronized between the peers.
  - **N3000:** vPC switch cannot be RP/MSDP peer - As we do not sync PIM states between vPC peers

# PIM RP Configuration

## Rendezvous Point

```
feature pim
ip pim rp-address 10.1.4.4 group-list 232.0.0.0/8
ip pim rp-address 10.1.4.4 prefix-list RP_Prefix
ip pim rp-address 10.1.4.4 route-map RP_rMAP
```

→ Static RP Configuration

```
feature pim
ip pim auto-rp forward listen
ip pim auto-rp mapping-agent loopback 0 scope 32
ip pim auto-rp rp-candidate loopback 0 group-list<>
```

→ Auto-RP Configuration

```
feature pim
ip pim bsr forward listen
ip pim bsr bsr-candidate loopback 0
ip pim bsr rp-candidate loopback 0 group-list<>
```

→ BSR Configuration

# PIM Anycast RP

- Primarily used for redundancy and load sharing.
  - Same address configured on different RP nodes.
- LHR forwards (\*,G) and joins the shared tree with the closest RP.
- FHR registers the source with the closest RP.
- State entries must be synchronized among the RPs.
  - MSDP
  - PIM

# PIM Anycast RP Configuration

## Sw1 Configuration

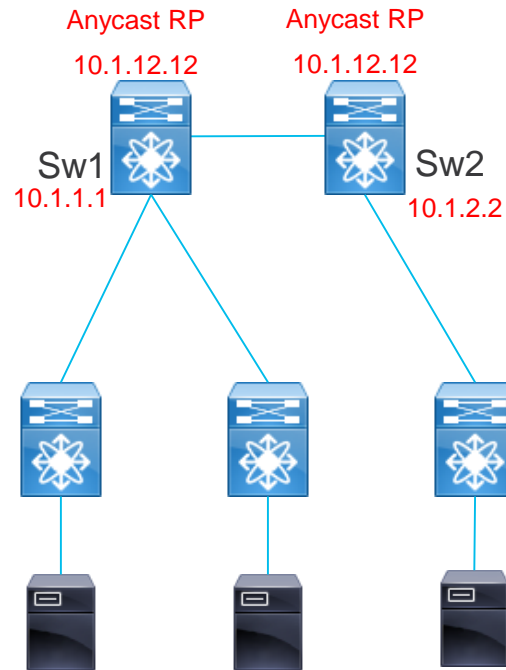
```
feature pim
ip pim rp-address 10.1.12.12 group-list 232.0.0.0/8
ip pim anycast-rp 10.1.12.12 10.1.1.1
ip pim anycast-rp 10.1.12.12 10.1.2.2
```

## Sw2 Configuration

```
feature pim
ip pim rp-address 10.1.12.12 group-list 232.0.0.0/8
ip pim anycast-rp 10.1.12.12 10.1.1.1
ip pim anycast-rp 10.1.12.12 10.1.2.2
```

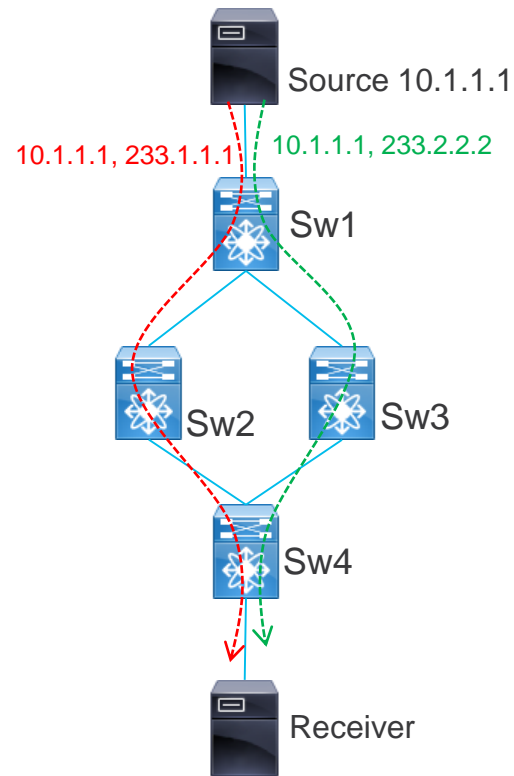
```
NxOS1# show ip pim rp
Anycast-RP 10.1.12.12 members:
 10.1.1.1* 10.1.2.2

RP: 10.1.12.12, (0), uptime: 00:01:23, expires: never,
priority: 255, RP-source: (local), group ranges:
224.0.0.0/4
NxOS1#
```



# Multicast Load Distribution

- Multicast load balancing over ECMP RPF interfaces.
- Enabled by default in NXOS
- Different options available
  - None
  - S,G Hashing
  - Resilience





# PIM BFD

- Nexus Platforms supports configuring PIM as BFD client.
- Rapid failure detection and protection.
- Enable globally for PIM, disable per interface if desired

```
feature bfd
ip pim bfd

interface vlan 100
 ip pim bfd-instance disable
```

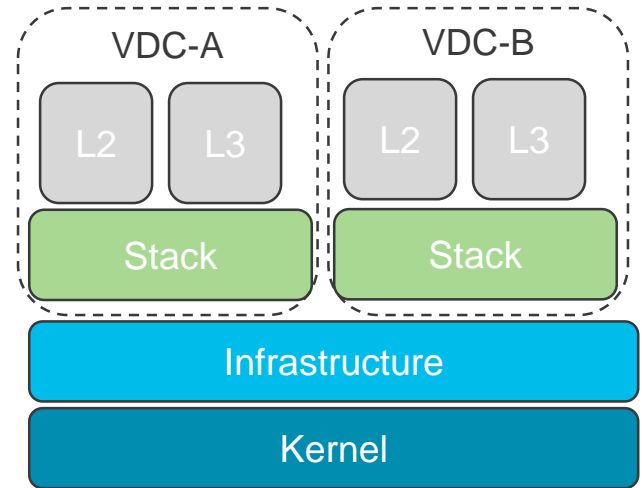
```
NxOS1# show ip pim neighbor
```

```
PIM Neighbor Status for VRF "default"
```

Neighbor	Interface	Uptime	Expires	DR Priority	Bidir- Capable	BFD State
192.168.100.1	Vlan100	07:39:51	00:01:16	1	yes	Up

# VDC Multicast Resource Allocation

- Virtual Device Context (VDC) enables control plane virtualization and shares the hardware.
- There are 3 types of VDC resource allocation:
  - Global Resource
  - Dedicated Resource
  - Shared Resource
- Control Plane process like PIM, IGMP will be dedicated to each VDC.
- Forwarding entries in hardware will be from shared resource.



```
vdc NXOS1
  limit-resource m4route-mem minimum <> maximum <>
```

# CoPP Interaction

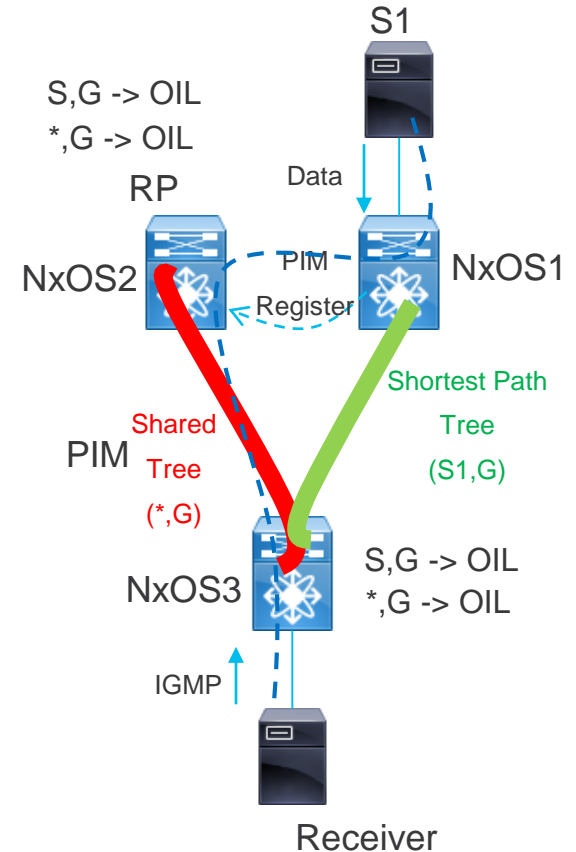
- Control Plane Policing (CoPP) is enabled by default in all Nexus Platforms.
- In most Nexus platforms, RPF failure will be punted to CPU at a very low rate.
- Nexus 9000 platforms will ALWAYS punt RPF failure to CPU
  - Helps to learn the multicast source information.
- Default values may not fit all environments.
- Tweaking the attributes (PIM, IGMP, MSDP etc) may need monitoring and adjustment.

# NXOS Multicast Forwarding

## State Creation and Packet forwarding

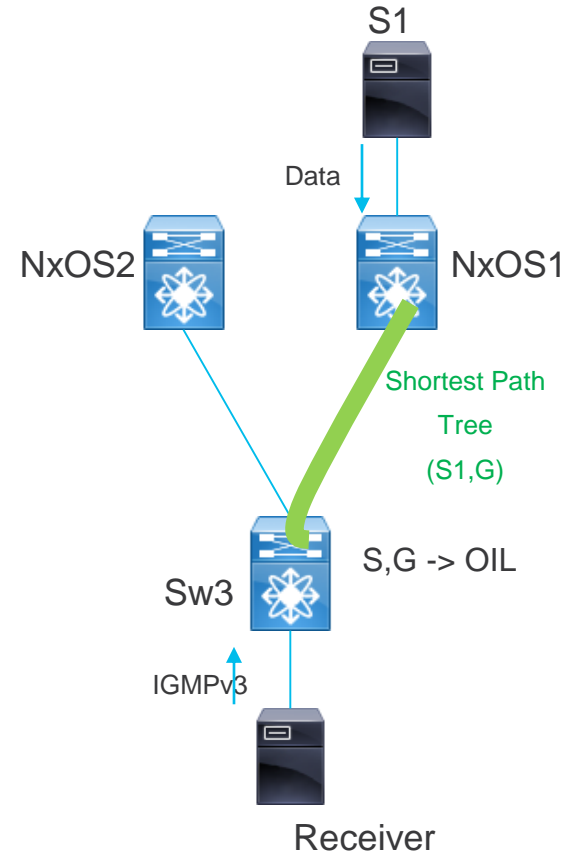
# PIM Sparse Mode States

- LHR creates (\*,G) upon receiving IGMP Join message from host.
  - It builds Shared tree towards RP
- RP creates (\*,G) entry upon receiving PIM join message from downstream node.
- FHR upon receiving the data traffic from source, will create (S,G) and registers with RP.
- RP creates (S,G) entry upon receiving PIM Register message from FHR.
  - RP forwards the data over shared tree
- LHR creates (S,G) upon receiving data traffic.
  - It builds shortest tree towards Source.



# PIM SSM mode States

- LHR creates (S,G) upon receiving IGMP Join message from host.
  - SSM requires IGMPv3 that carries both S,G
- LHR builds the tree towards source by sending PIM Join message.
- FHR creates (S,G) upon receiving PIM join message.



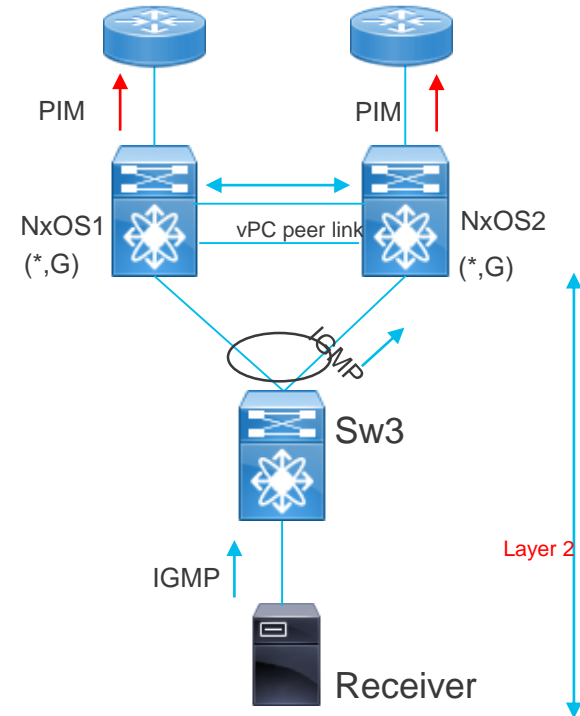
# PIM Bidir mode States

- State creation are similar to Shared tree in PIM Sparse mode .
- LHR will create (\*,G) upon receiving IGMP join message.
  - It send PIM join towards RP.
- On upstream routers, PIM joins create (\*,G) state all the way to the RP
- On source-only branches, control plane installs (\*,G/m) entries to enable data forwarding toward bidir RP

# Multicast States and Packet Flow in vPC environment

## vPC LHR

- Receiver sends IGMP join towards vPC peer.
  - Sw3 updates the Snooping table with (\*,G)
  - Forwards the IGMP Join message to NxOS2
- NxOS2 creates (\*,G) entry and sync the state with NxOS1.
  - NxOS2 sends IGMP packet to NxOS1 by encapsulating it in CFS message
- NxOS1 creates (\*,G) entry in local table.
- Both NxOS1 and NxOS2 builds shared tree towards RP.





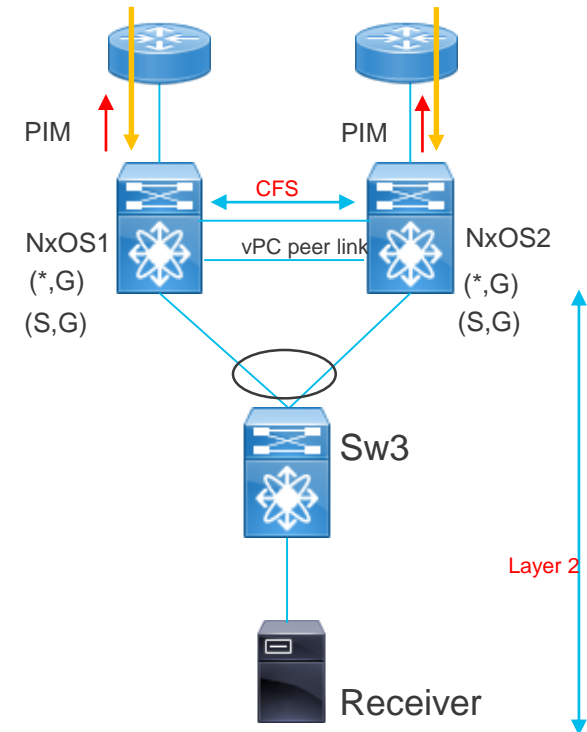
# Multicast States and Packet Flow in vPC environment

## vPC LHR

```
NxOS1# show ip mroute 233.1.1.1
IP Multicast Routing Table for VRF "default"
(*, 233.1.1.1/32), uptime: 00:39:35, igmp ip pim
  Incoming interface: Ethernet1/1, RPF nbr: 10.1.14.4
  Outgoing interface list: (count: 1)
    Vlan100, uptime: 00:39:35, igmp
NxOS1#
```

(\*,G) built from both peers

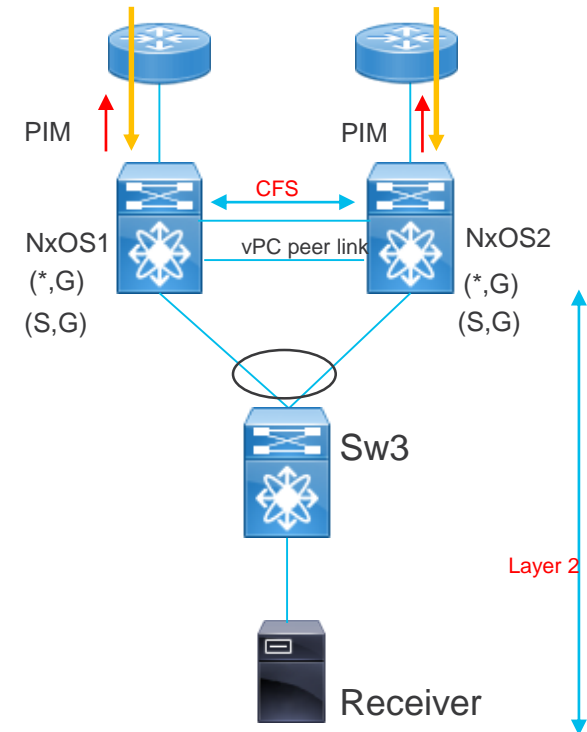
```
NxOS2# show ip mroute 233.1.1.1
IP Multicast Routing Table for VRF "default"
(*, 233.1.1.1/32), uptime: 00:41:07, igmp ip pim
  Incoming interface: Ethernet1/1, RPF nbr: 10.1.25.5
  Outgoing interface list: (count: 1)
    Vlan100, uptime: 00:41:07, igmp
NxOS2#
```



# Multicast States and Packet Flow in vPC environment

## vPC LHR

- Both vPC peers receives data traffic over shared tree from RP.
  - $(*,G)$  was built from both peers towards RP.
- Peers build  $(S,G)$  towards the source.
- Peers negotiate for Active Forwarder Role.
  - CFS messages exchanged with metric to reach the source S.
- Winner continues forwarding the traffic. The other peer will remove OIF and stop tx PIM Joins.



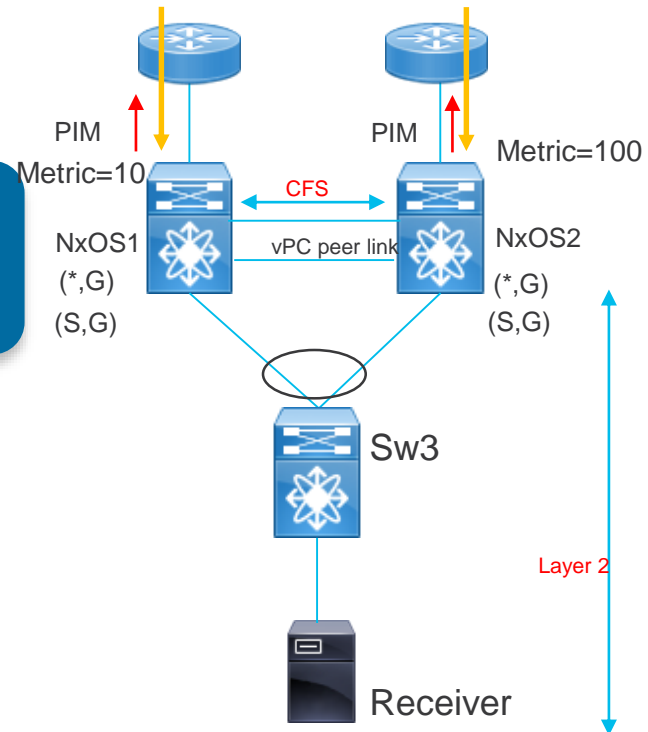
# Multicast States and Packet Flow in vPC environment

## vPC LHR

```
NxOS1# show ip mroute 233.1.1.1
IP Multicast Routing Table for VRF "default"
<removed>
(10.1.7.7/32, 233.1.1.1/32), uptime: 00:00:04, ip mrib pim
  Incoming interface: Ethernet1/1, RPF nbr: 10.1.14.4
  Outgoing interface list: (count: 1)
    Vlan100, uptime: 00:00:04, mrib
NxOS1#
```

```
NxOS2# show ip mroute 233.1.1.1
IP Multicast Routing Table for VRF "default"
<removed>
(10.1.7.7/32, 233.1.1.1/32), uptime: 00:00:03, ip mrib pim
  Incoming interface: Ethernet1/1, RPF nbr: 10.1.25.5
  Outgoing interface list: (count: 0)
NxOS2#
```

OIL is  
populated in  
Active  
Forwarder



# Multicast States and Packet Flow in vPC environment

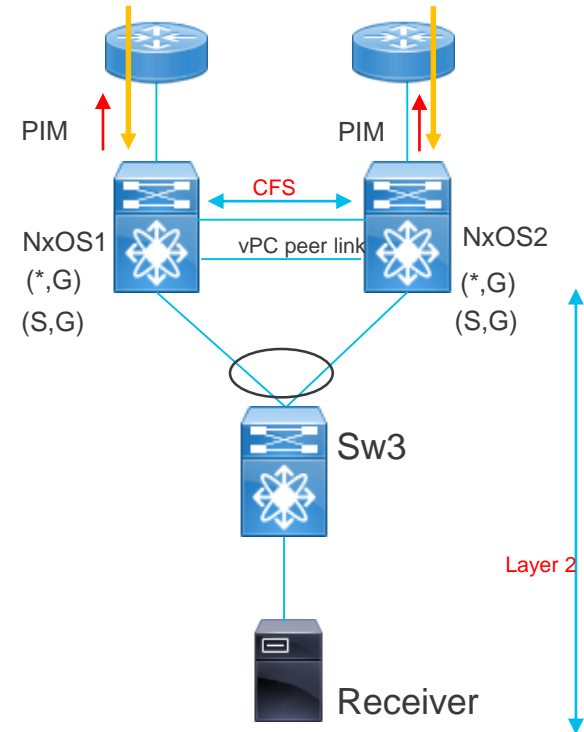
## Pre-Built SPT

```
ip pim pre-build-spt
```

- Potential Optimization for faster convergence
- The difference is – Both peers continuously sends PIM Joins.
- Active Forwarder adds OIL and forwards the packet.
- The other peer will not populate the OIL.

### Pre-built SPT Considerations:

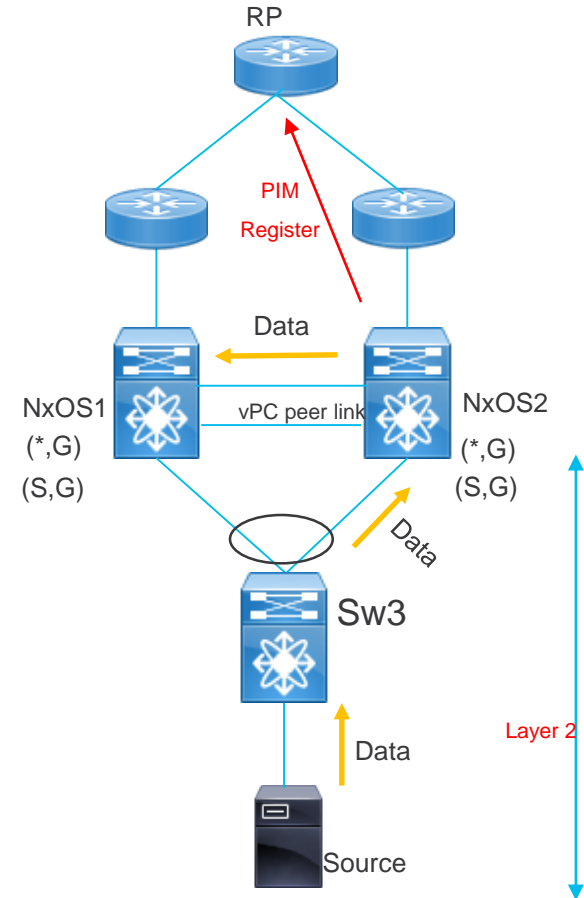
- Improves re-convergence time
- Consumes additional bandwidth



# Multicast States and Packet Flow in vPC environment

## vPC FHR

- Sw3 receives source traffic and forward to NxOS2 based on ether channel hashing.
- NxOS2 will forward the data traffic over vPC peer-link.
  - vPC peer link will be marked as mrouter port.
- Both peers create (S,G) state entry.
- PIM DR will register the source with RP.



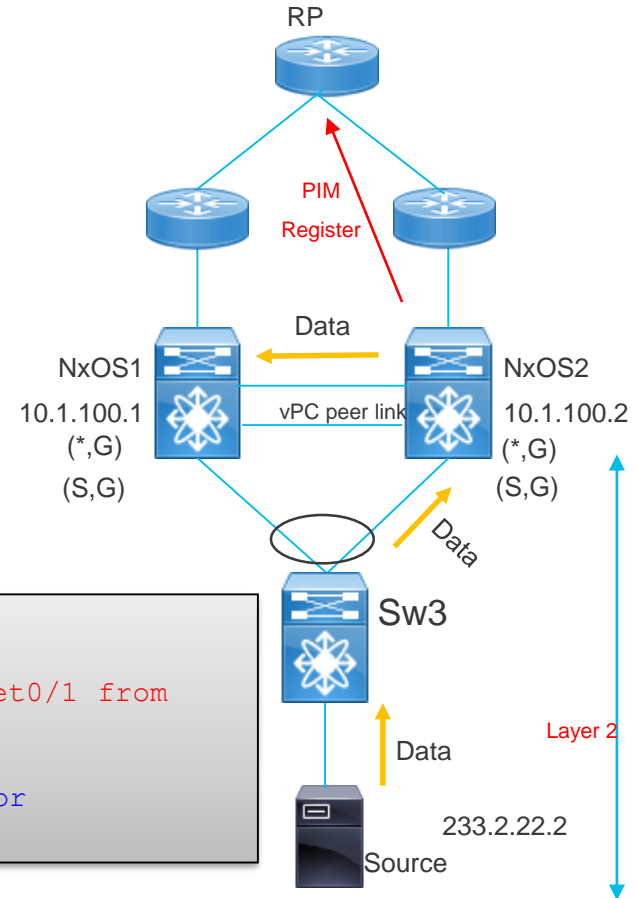
# Multicast States and Packet Flow in vPC environment

## vPC FHR

```
NxOS1# show ip mroute 233.2.22.2
IP Multicast Routing Table for VRF "default"
(10.1.100.5/32, 233.2.22.2/32), uptime: 00:04:28, ip pim
  Incoming interface: Vlan100, RPF nbr: 10.1.100.5
  Outgoing interface list: (count: 0)
N9kv-1#
```

```
NxOS2# show ip mroute 233.2.22.2
IP Multicast Routing Table for VRF "default"
(10.1.100.5/32, 233.2.22.2/32), uptime: 00:03:19, ip pim
  Incoming interface: Vlan100, RPF nbr: 10.1.100.5
  Outgoing interface list: (count: 0)
N9kv-2#
```

```
RP#
RP#show logg
*Jan 21 01:07:16.756: PIM(0): Received v2 Register on GigabitEthernet0/1 from
10.1.100.2
*Jan 21 01:07:16.756:      for 10.1.100.5, group 233.2.22.2
*Jan 21 01:07:16.756: PIM(0): Send v2 Register-Stop to 10.1.100.2 for
10.1.100.5, group 233.2.22.2
```

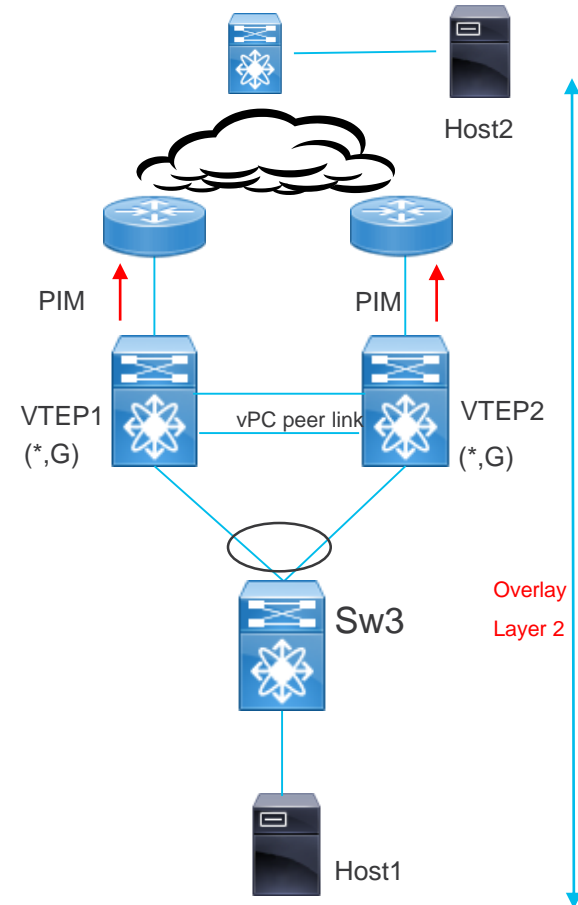


# Multicast States and Packet Flow in VxLAN environment

## vPC VTEPs

```
interface nve1
 member vni 100100 mcast-group 232.1.1.1
```

- Procedure is similar to Sparse mode state creation.
- Configuration driven and not IGMP.
- NVE configuration will trigger building shared tree towards RP.
- For each (S,G) one of VTEP will be the active forwarder.
- BUM traffic received by active forwarder will be flooded to Layer 2 network.



# NXOS Multicast Platform Independent Troubleshooting: Component Responsibilities and Interactions



# Component Responsibilities and Interactions

## Event Histories

- **Event-histories** - are “running debugs” **on by default**.
- No more waiting for maintenance windows to debug 😊
- SG state creation (from data packet) example...

```
Nexus# show system internal mfwfwd event-history pkt
2017 Nov 23 15:44:31.973216 mcastfwd [9725]: [9767]: Create state for (100.4.21.101, 225.4.21.1)
```

- IGMP → MRIB Update with new OIF example...

```
Nexus# show ip igmp internal event-history debugs
2017 Nov 23 15:44:26.356601 igmp [30223]: : Received v2 Report for 225.4.21.1 from 100.1.20.102 (Vlan120)

Nexus# show ip igmp internal event-history igmp-internal
2017 Nov 23 15:44:26.356687 igmp [30223]: [30389]: Inserting IGMP add-update for (*, 225.4.21.1) [i/f Vlan120] into MRIB buffer
2017 Nov 23 15:44:26.358551 igmp [30223]: [30382]: MRIB: Process route (*, 225.4.21.1) add vpc svi oif Vlan120
2017 Nov 23 15:44:26.358657 igmp [30223]: [30382]: MRIB: Added route (*,225.4.21.1) to tree
```

# Verifying NXOS Multicast Programming Debug Logfile

- Debug commands still available.
- Can be re-directed to files for more versatile use.

```
Nexus# debug logfile mcast-pending
Nexus# dir log: | eg -i mcast
      0      Nov 23 15:19:27 2017  mcast-pending
```

1) Creates a “debug log” file

```
Nexus# debug ip pim internal
Nexus# debug mfdm event
Nexus# debug mfdm error
```

2) Enables desired debugs

```
Nexus# dir log: | eg -i mcast
 23870      Nov 23 15:25:01 2017  mcast-pending
```

3) Shows contents of “debug log” file

```
Nexus# show debug logfile mcast-pending
2017 Nov 23 15:24:19.678834 pim: [10426] (100.3.20.101/32, 225.131.38.2 /32) expiration timer updated due to data activity
2017 Nov 23 15:24:19.679216 pim: [10426] (200.5.1.70/32, 225.5.1.50/32) expiration timer updated due to data activity
2017 Nov 23 15:24:19.681404 pim: [10426] For RPF Source 100.3.20.101 RPF neighbor 172.16.200.1 and iif Ethernet2/13
```

# Component Responsibilities and Interactions Cheat Sheet



- Individual **tech-supports** and **event histories**
- **MRIB**
  - `show ip mroute`  
`show tech-support routing ip multicast`
  - `Show tech-support multicast`  
`show routing ip multicast event-history <variables>`
- **PIM**
  - `show ip pim route`  
`show tech-support ip pim`  
`show ip pim event-history <variables>`
- **IGMP**
  - `show ip igmp route`  
`show tech-support ip igmp`  
`show ip igmp event-history <variables>`  
`show tech-support ip igmp snooping`  
`show ip igmp snooping event-history <variables>`

# Component Responsibilities and Interactions

## Cheat Sheet cont...



- **MFWD** (Multicast Forwarding - on by default)
  - `show tech-support mfwd`  
`show system internal mfwd event-history <variables>`
- **MFDM** (Multicast Forwarding Distribution Module - on by default)
  - `show tech-support routing ip multicast`  
`show system internal mfdm event-history <variables>`
- Most versions of software will now bundle most multicast components into a **single tech-support.**
  - `sh tech-support ip multicast | eg tech-support`
    - ``show tech-support ip igmp``
    - ``show tech-support ip igmp snooping``
    - ``show tech-support mfwd``
    - ``show tech-support routing ip multicast``
    - ``show tech-support ip pim``

# Component Responsibilities and Interactions

## Cheat Sheet



- Individual **tech-supports** and **event histories**
- **M2RIB (VxLAN Scenarios)**
  - `show l2 mroute`
  - `show tech-support multicast-vxlan-evpn`
- **MFDM (VxLAN Scenarios)**
  - `show forwarding distribution l2 multicast`  
`show forwarding distribution ip igmp snooping`
  - `show forwarding distribution multicast outgoing-interface-list l2`
- **Show tech**
  - Most of the show-techs defined earlier are useful

# Verifying NXOS Multicast Programming Short on Time - Cheat Sheet



- No time to debug or troubleshoot?
- Try to gather the following before clearing or resolving any issue.
  - a) Working and Non-Working Group/OIF (for comparison)
  - b) Timestamp of any related events if applicable
    - `show logging logfile` and `'show accounting log'` are helpful
  - c) `show-tech` from system **as well as peer system** (vPC or non-vPC).
    - `show tech-support ip multicast > bootflash:<switchname>-tech-ipmc`
    - `show tech-support m2rib > bootflash:<switchname>tech-m2rib`
    - `show tech-support m2fib > bootflash:<switchname>tech-m2fib`
    - `show tech-support forwarding l2 multicast > bootflash:<switchname>tech-l2mcast`
    - `show tech-support forwarding l3 multicast > bootflash:<switchname>tech-l3mcast`
    - `show tech-support routing ip unicast detail > bootflash:<switchname>tech-ucast`
    - `Show tech-support multicast-vxlan-evpn`

Shows all CLI run on box  
&  
Survives reloads

# NXOS Multicast Platform Independent Troubleshooting: State Creation and Packet forwarding

# Verifying NXOS Multicast Programming

## First Packet Processing

- **Problem Symptom:** *“We seem to be dropping the first packet(s) at the start of our multicast streams”*
- NXOS devices **do not forward packets in software** by default.
- This is done to not overwhelm CPU in large scale mcast environments.
- Can be toggled via CLI `ip routing multicast software-replicate`
  - Not recommended to change default – increased CPU impact.



# Verifying NXOS Multicast Programming

## Other CPU Packets

- Data driven state entry creations.
- Data Packets from connected source on FHR
  - Creates (S,G) state on FHR
  - Trigger PIM RP-registration
- PIM Register packets from FHR to RP
- RPF-Fail packets (at intervals)
- Data-Packets at LHR to create (S,G) state and trigger SPT switchover.
- Typical control plane packets (PIM, IGMP, MSDP, etc...)
- You can utilize **Ethalyzer** on Nexus devices to verify / identify CPU packets...

# Verifying NXOS Multicast Programming Ethalyzer

- Ethalyzer is a built-in sniffer that can capture traffic to/from the CPU.
- Utilize Ethalyzer to verify **CPU tx/rx** packets
- Removes requirement of external Sniffer attached
- Only **the first packet(s)** should be punted to Software...

```
Nexus# ethalyzer local interface inband capture-filter "host 225.4.21.1"  
2017-11-23 20:25:02.007844 100.4.21.101 -> 225.4.21.1 IP Unknown (0xfd)  
2017-11-23 20:25:02.008862 100.4.21.101 -> 225.4.21.1 IP Unknown (0xfd)
```

- PIM Join / Prune Capture Example

```
Nexus# ethalyzer local interface inband capture-filter "src 172.16.200.5 and ip proto 103"  
2017-11-22 11:29:34.046511 172.16.200.5 -> 224.0.0.13 PIMv2 Hello  
2017-11-22 11:29:59.966210 172.16.200.5 -> 224.0.0.13 PIMv2 Join/Prune
```

# Verifying NXOS Multicast Programming

## Reading Mroute Table

```
Nexus# show ip mroute
(*, 235.50.0.1/32), uptime: 1w0d, igmp ip pim
  Incoming interface: Ethernet2/13, RPF nbr: 172.16.200.1
  Outgoing interface list: (count: 2)
    Vlan501, uptime: 1w0d, igmp
    Vlan500, uptime: 1w0d, igmp

(200.50.0.100/32, 235.50.0.1/32), uptime: 1w0d, pim ip
  Incoming interface: Vlan500, RPF nbr: 200.50.0.100
  Outgoing interface list: (count: 3)
    Vlan501, uptime: 1w0d, mrrib
    Vlan500, uptime: 1w0d, mrrib, (RPF)
    Ethernet2/13, uptime: 1w0d, pim
```

This output is from  
Supervisor  
Perspective (MRIB)

- **IGMP**: (\*,G) Entry (and OIF) populated via IGMP join
- **PIM (OIF)**: PIM populated this OIF
- **PIM (\*,G & S,G)**: Confirms MRIB -> PIM communication (for PIM joins)
- **MRIB**: (\*,G) OIF copied to (S,G) entry

# Verifying NXOS Multicast Programming

## Mroute Flags

- Wait, where are the flags from IOS?
  - Flags and similar info are still present...

```
Nexus# show forwarding multicast route
```

```
Legend:
```

```
C = Control Route  
D = Drop Route  
G = Local Group (directly connected receivers)  
O = Drop on RPF failure  
P = Punt to Supervisor  
W = Wildcard  
d = Decap route
```

These flags appear in “show forwarding...” commands

```
Nexus# show ip pim route internal
```

```
(*, 235.50.0.1/32), RP 200.200.200.200, expires 00:00:30, RP-bit  
Incoming interface: Ethernet2/13, RPF nbr 172.16.200.1  
RPF-Source 200.200.200.200, JP-holdtime 180, [0/0]  
<snip>
```

Flags in client tables as well

# Verifying NXOS Multicast Programming Pending Routes

- **Problem Symptom:** *Why does my mroute say “pending” ?*

```
Nexus# show ip mroute
(200.50.0.100/32, 235.50.0.1/32) [pending], uptime: 30w3d, ip mrib pim
Incoming interface: Vlan500,, RPF nbr: 200.50.0.100,
Outgoing interface list: (count: 1)
  Vlan501, uptime: 3d01h, mrib
```

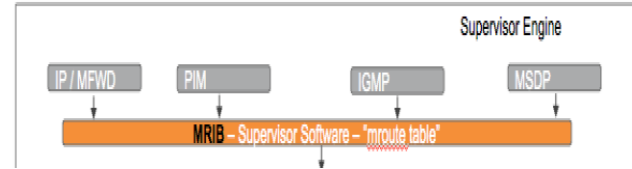
- Routes **not yet programmed in hardware** will show “Pending”
- Pending routes ‘*could*’ be temporary (seconds), but should not be permanent.
- Typically result of issue between **Software and Hardware programming**

If all mroutes start at same time with large enough scale

Let's verify stages of mroute programming...

# Verifying NXOS Multicast Programming

## Mroute Programming: Supervisor



1. Troubleshooting should start with MRIB client tables
  - PIM and IGMP client “route table” examples

```
Nexus# show ip pim route
(200.50.0.100/32, 235.50.0.1/32), expires 00:03:00
  Incoming interface: Vlan500, RPF nbr 200.50.0.100
  Oif-list:          (1) 00000000, timeout-list: (0) 00000000
  Timeout-interval: 1, JP-holdtime round-up: 3
```

Expiration timer

ONLY shows PIM  
oifs, not IGMP oifs

```
Nexus# show ip igmp route
IGMP Connected Group Membership for VRF "default" - 3 total entries
Type: S - Static, D - Dynamic, L - Local, T - SSM Translated
Group Address      Type Interface      Uptime    Expires    Last Reporter
235.50.0.1         D   Vlan500           1w1d     00:03:04   200.50.0.200
235.50.0.1         D   Vlan501           1w1d     00:03:13   200.50.1.200
```

Similar to IOS  
tables

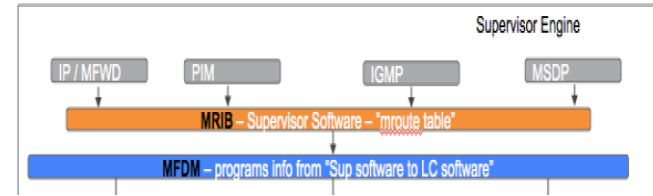
- If Mroute & Client tables are correct, let's confirm next stage of programming...

# Verifying NXOS Multicast Programming

## Mroute Programming: MFDM

### 2. Confirm MRIB has updated MFDM with mroute

- MRIB (supervisor) → MFDM (supervisor)



```
Nexus# show forwarding distribution multicast route  
<snip>
```

```
(*, 235.50.0.1/32), RPF Interface: Ethernet2/13, flags: G  
Received Packets: 0 Bytes: 0  
Number of Outgoing Interfaces: 2  
Outgoing Interface List Index: 23  
Vlan500  
Vlan501
```

Confirm (\*,G) and respective OIFs are present, and Note any Flags that are set

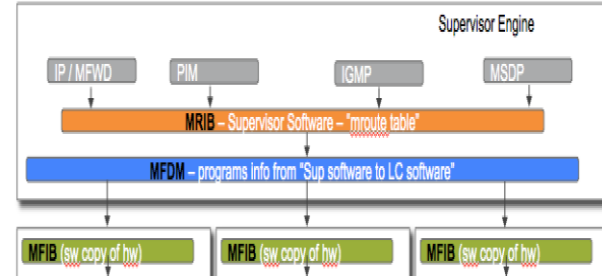
```
(200.50.0.100/32, 235.50.0.1/32), RPF Interface: Vlan500, flags:  
Received Packets: 82620035 Bytes: 82289554860  
Number of Outgoing Interfaces: 1  
Outgoing Interface List Index: 22  
Vlan501
```

RPF OIF will not appear here, as no re-write required.

# Verifying NXOS Multicast Programming

## Mroute Programming: MFIB

- Confirm Forwarding Engine (FE) software level.
  - MRIB (sup sw) → MFDM (sup sw) → **MFIB (FE sw)**
  - MFIB is the Forwarding Engine's **software perspective**.



```
Nexus# show forwarding ip multicast route
slot 2
=====
<snip>
(200.50.0.100/32, 235.50.0.1/32), RPF Interface: Vlan500, flags:
  Received Packets: 66028409868 Bytes: 8187522823632
  Number of Outgoing Interfaces: 1
  Outgoing Interface List Index: 22
  Vlan501 Outgoing Packets:1905639494 Bytes:236299291250
```

Per module info

Verify BOTH ingress & egress

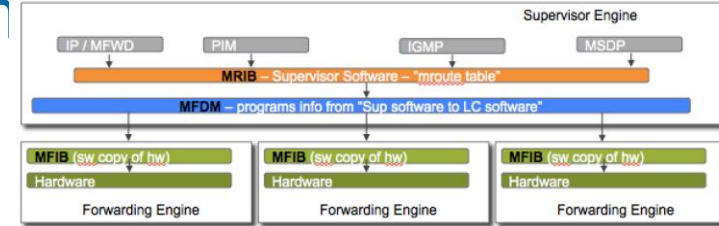
Verify SG, RPF and OIF



# Verifying NXOS Multicast Program Mroute Programming: Hardware

## 4. Final step: Forwarding Engine (FE) hw level.

- MRIB (sup sw) → MFDM (sup sw) → MFIB (FE sw) → Fwd Engine (hw)



```
Nexus# show system internal forwarding multicast route detail
slot 2
=====
(200.50.0.100/32, 235.50.0.1/32), Flags: *S
Lamira: 1, HWIndex: 0x1a8c, VPN: 1
RPF Interface: Vlan500, LIF: 0x47, PD oiflist index: 0x1
ML3 Adj Idx: 0xa00b, MD: 0x2001, MET0: 0x0, MET1: 0x2002, MTU Idx: 0x1
Rewrite Instance: 0
Dev: 1 Index: 0xa012   Type: MDT       elif: 0xc0002
                        dest idx: 0x7bc6   recirc-dti: 0xe20000
Dev: 1 Index: 0x6cfb   Type: OIF       elif: 0x80cfb    Vlan502
                        dest idx: 0x0      smac: 0022.557a.7441
```

Keys to Verify

Output may differ slightly  
per platform since this is  
HW perspective

- The (S,G) is present
- It's present for any ingress or egress module the traffic traverses
- The RPF information is correct

Let TAC and myself bang our heads with all the other "special indexes"

# Verifying NXOS Multicast Programming Command Cheat Sheet



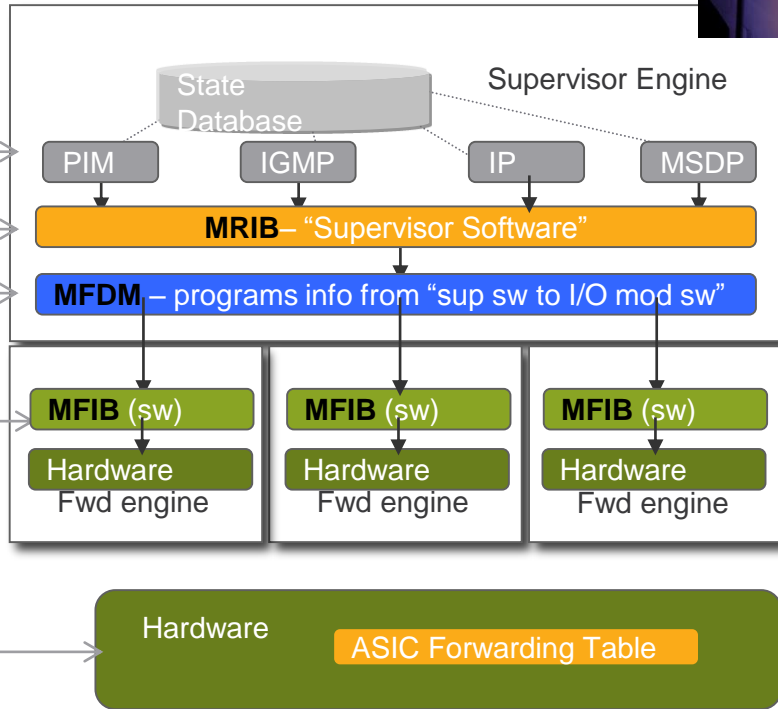
show ip pim route  
show ip igmp route  
show ip igmp snooping groups  
show ip msdp route

show ip mroute (show routing ip multicast)

show forwarding distribution ip multicast route  
show forwarding distribution ip igmp snooping

show forwarding ip multicast route

show system internal forwarding ip multicast route  
show system internal ip igmp snooping



# NXOS Multicast Platform Independent Troubleshooting: vPC Environment

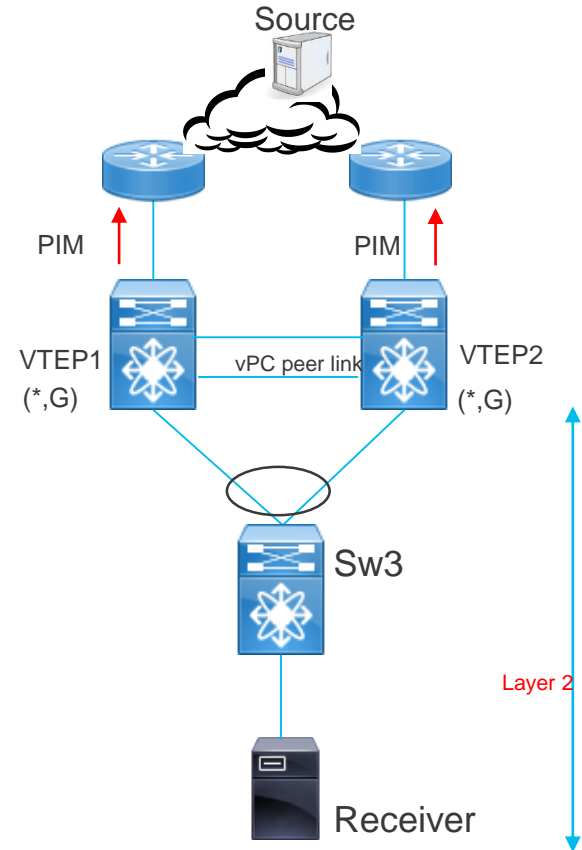
# Troubleshooting Multicast in vPC Environments

## Ingress L3, Egress vPC

### Problem Symptom:

What to do If vPC hosts are not receiving the multicast ?

- L3 Sources
- Receivers are connected to vPC Peers
- vPC vlan defined as any vlan that is configured on vPC peer-link



# Troubleshooting Multicast in vPC Environments

## L3 → vPC: Initial Join

### 1. (\*,G) State creation in vPC via IGMP Join

- a. Receiver sends IGMP join,
- b. Creates: (1) snooping, (2) IGMP, and (3) \*,G state with vPC link as OIF

```
Nexus-2# show ip igmp snooping groups 225.131.38.2 | exc */*
```

Type: S - Static, D - Dynamic, R - Router port

Vlan	Group Address	Ver	Type	Port list
143	225.131.38.2	v2	D	Po1

```
Nexus-2# show ip igmp route 225.131.38.2
```

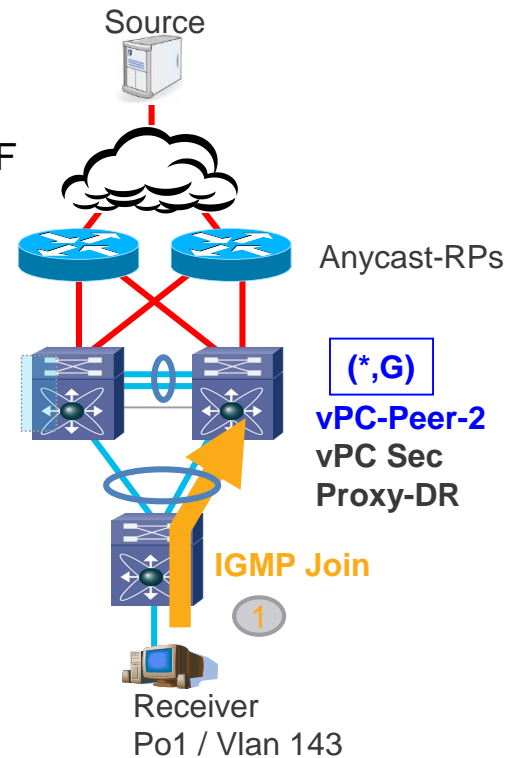
Type: S - Static, D - Dynamic, L - Local, T - SSM Translated

Group Address	Type	Interface	Uptime	Expires	Last Reporter
225.131.38.2	D	Vlan143	00:00:47	00:03:32	100.111.43.3

```
Nexus-2# show ip mroute 225.131.38.2
```

IP Multicast Routing Table for VRF "default"

```
(*, 225.131.38.2/32), uptime: 00:00:57, igmp ip pim
Incoming interface: Ethernet2/13, RPF nbr: 172.25.250.1
Outgoing interface list: (count: 1)
Vlan143, uptime: 00:00:57, igmp
```



# Troubleshooting Multicast in vPC Environments

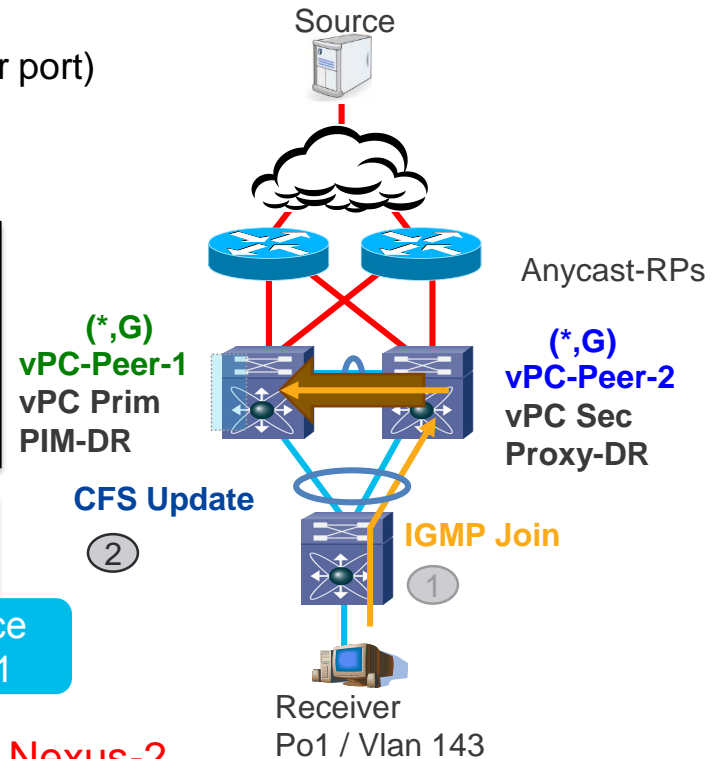
## L3 → vPC: IGMP Sync

2. Peer-2 notifies Peer-1 via CFS. (cisco fabric services)
  - a. Native packet also forwarded across vPC peer-link (mrouter port) creating **(\*,G)** on vPC-Peer1
  - b. You can confirm CFS communication

```
Nexus-2# show ip igmp snooping internal event-history vpc
vPC Events for IGMP Snoop process
2017 Feb  911:49:07.021296 igmp [29690]: : Sent
IGMP_SNOOP_vPC_IGMP_PACKET to peer
2017 Feb  911:49:07.021283 igmp [29690]: : Doing CFS unreliable send
2017 Feb  911:49:07.021210 igmp [29690]: : Send IGMP PACKET to peer
over CFS
```

```
Nexus-1# show ip igmp snooping internal event-history vpc
vPC Events for IGMP Snoop process
2017 Feb  911:49:07.022928 igmp [4585]: : Received a CFS message with
MCEC SI 1
2017 Feb  911:49:07.022919 igmp [4585]: : Received a CFS
```

“SI 1” = source interface Po1



This should result in Nexus-1 creating exact same state as Nexus-2...

# Troubleshooting Multicast in vPC Environments

## L3 → vPC: IGMP Sync cont...

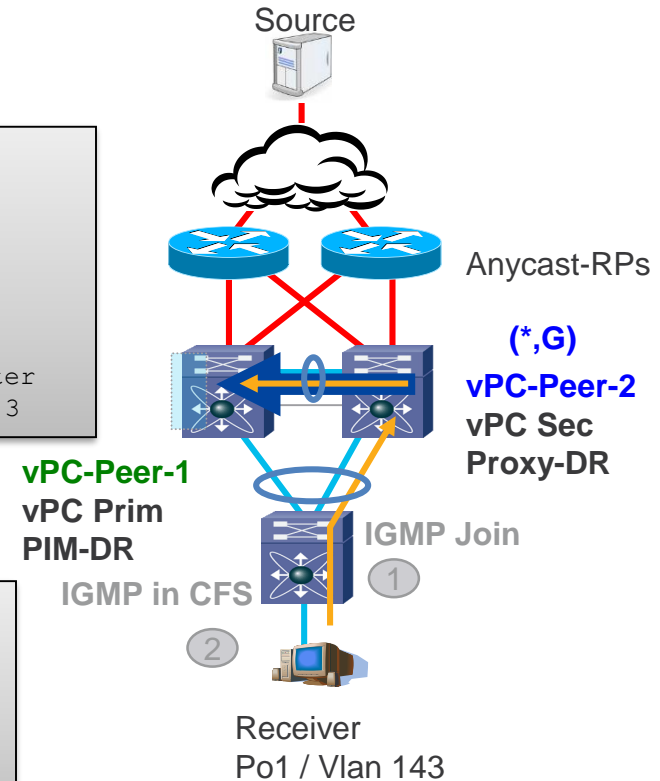
c. Verify IGMP and (\*,G) state on vPC-Peer 1

```
Nexus-1# show ip igmp snooping groups 225.131.38.2 | exc */*  
Type: S - Static, D - Dynamic, R - Router port  
Vlan Group Address Ver Type Port list  
143 225.131.38.2 v2 D Po1
```

```
Nexus-1# show ip igmp route 225.131.38.2  
Type: S - Static, D - Dynamic, L - Local, T - SSM Translated  
Group Address Type Interface Uptime Expires Last Reporter  
225.131.38.2 D Vlan143 00:00:57 00:03:22 100.111.43.3
```

```
Nexus-1# show ip mroute 225.131.38.2  
IP Multicast Routing Table for VRF "default"
```

```
(* , 225.131.38.2/32), uptime: 00:01:03, igmp ip pim  
Incoming interface: Ethernet2/13, RPF nbr: 172.25.250.1  
Outgoing interface list: (count: 1)  
Vlan143, uptime: 00:01:03, igmp
```



# Troubleshooting Multicast in vPC Environments

## L3 → vPC: RPF Forwarder

Both vPC peers should have same \*,G state

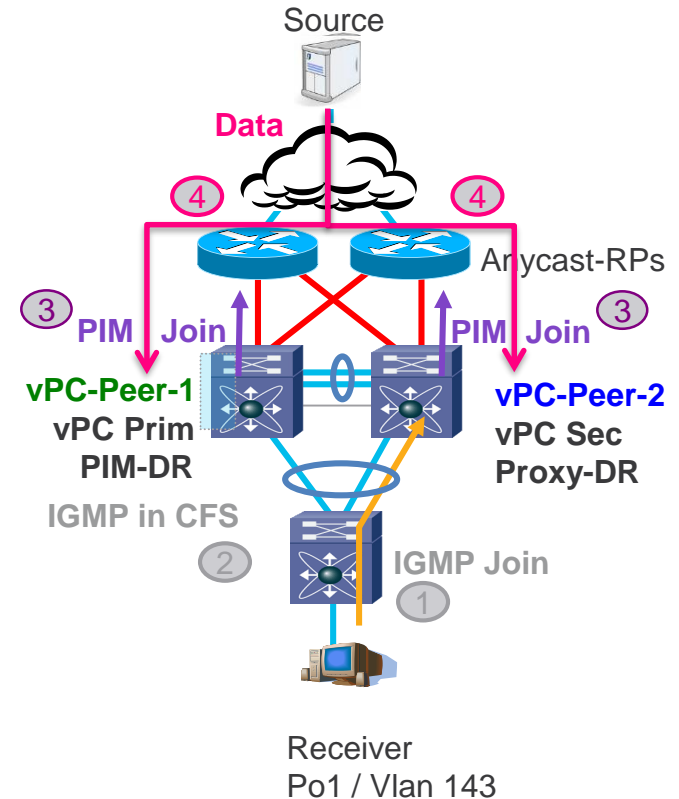
3. Both vPC peers tx PIM Joins

4. Both vPC Peers rx the multicast, and create SG state.

```
Nexus-1# show ip mroute
(100.3.20.101/32, 225.131.38.2/32), uptime: 00:00:01, ip pim
Incoming interface: Ethernet2/13, RPF nbr: 172.16.200.1
Outgoing interface list: (count: 1)
    Vlan143, uptime: 00:00:01, igmp
```

```
Nexus-2# show ip mroute
(100.3.20.101/32, 225.131.38.2/32), uptime: 00:00:04, ip pim
Incoming interface: Ethernet2/13, RPF nbr: 172.16.200.5
Outgoing interface list: (count: 1)
    Vlan143, uptime: 00:00:01, igmp
```

Temporary duplicates at while creating SG state until...





# Troubleshooting Multicast in vPC Environments

## L3 → vPC: RPF Forwarder

### 5. “RPF-Forwarder” is determined to prevent duplicates

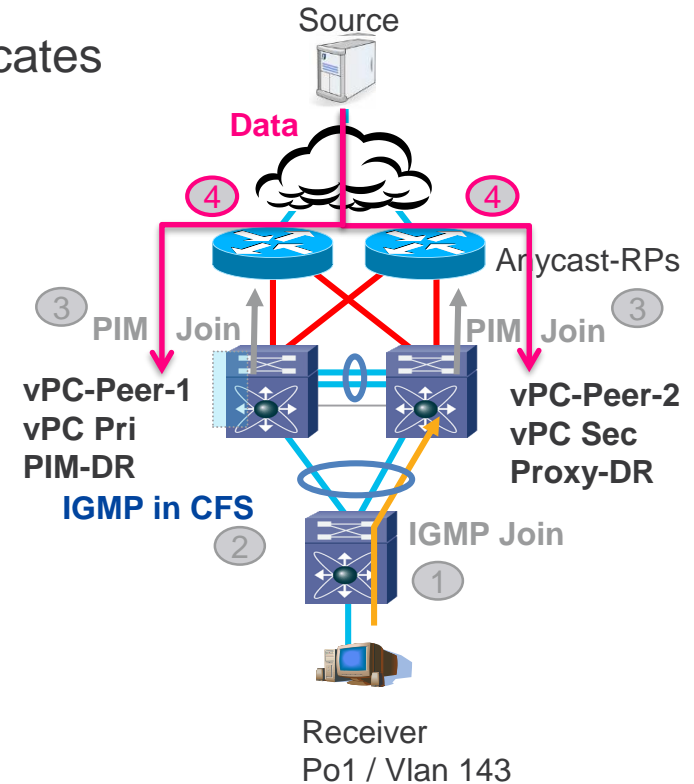
- Best RPF metric to source = win
- in case of tie... vPC Primary wins

```
Nexus-1# show ip pim internal vpc rpf-source
```

```
Source: 100.3.20.101  
Pref/Metric: 110/5  
Source role: primary  
Forwarding state: Tie (forwarding)
```

```
Nexus-2# show ip pim internal vpc rpf-source
```

```
Source: 100.3.20.101  
Pref/Metric: 110/5  
Source role: secondary  
Forwarding state: Tie (not forwarding)
```



# Troubleshooting Multicast in vPC Environments

## L3 → vPC: RPF Forwarder

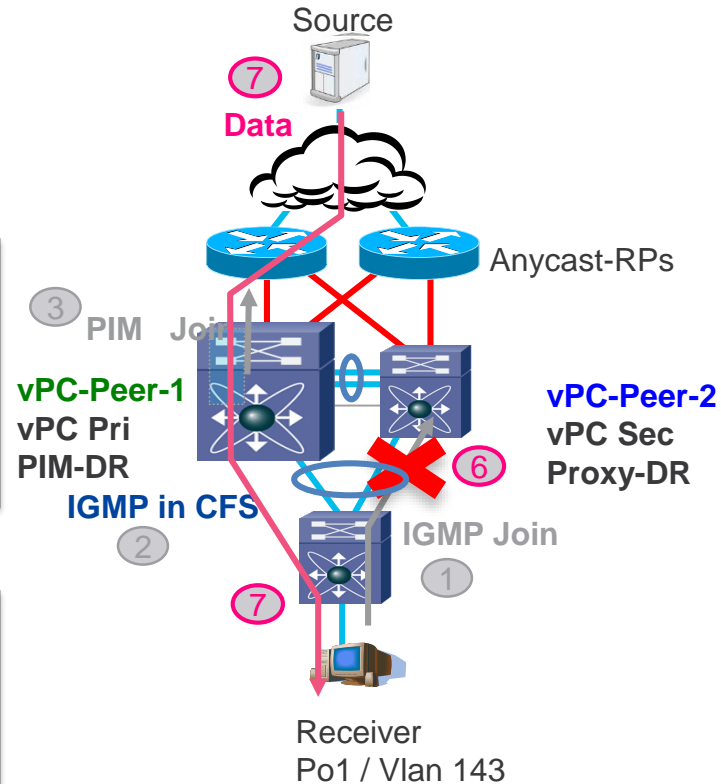
6. Loser removes OIFs & stop tx PIM Joins
7. Winner continues forwarding & tx PIM Joins

```
Nexus-1# show ip mroute
```

```
(100.3.20.101/32, 225.131.38.2 /32), uptime: 00:2:07, ip pim  
mrib  
Incoming interface: Ethernet2/13, RPF nbr: 172.16.200.1  
Outgoing interface list: (count: 1)  
Vlan143, uptime: 00:11:07, mrib
```

```
Nexus-2# show ip mroute
```

```
(100.3.20.101/32, 225.131.38.2 /32), uptime: 00:02:15, ip pim  
Incoming interface: Ethernet2/13, RPF nbr: 172.16.200.5  
Outgoing interface list: (count: 0)
```



# Troubleshooting Multicast in vPC Environments

## L3 → vPC: RPF Forwarder

- RPF Forwarder troubleshooting output

`show ip pim event-history vpc` or `debug ip pim vpc detail` (provide same output)

```
Nexus-1# show ip pim event-history vpc
```

```
2014 Feb  920:58:32.841524 pim: vPC: Send CFS RPF-source metric request for 1 sources
2014 Feb  920:58:32.841555 pim: vPC: Preparing CFS packet
2014 Feb  920:58:32.844674 pim: vPC: Received a CFS message
2014 Feb  920:58:32.844720 pim: vPC: Received RPF_source metric RESPONSE message with 1 Source-entries
2014 Feb  920:58:32.844824 pim: vPC: Processing RPF-source exchange message for source 100.3.20.101
2014 Feb  920:58:32.844962 pim: vPC: We win, our pref/metric: 110/5, peer's pref/metric: 110/5, adding all (S,G)-oifs to MRIB for source 100.3.20.101
```

```
Nexus-2# show ip pim event-history vpc
```

```
2014 Feb  920:58:32.843063 pim: vPC: Received a CFS message
2014 Feb  920:58:32.843092 pim: vPC: Received RPF_source metric REQUEST message with 1 Source-entries
2014 Feb  920:58:32.843201 pim: vPC: Processing RPF-source exchange message for source 100.3.20.101
2014 Feb  920:58:32.843305 pim: vPC: We lose, our pref/metric: 110/5, peer's pref/metric: 110/5, removing all (S,G)-oifs from MRIB for source 100.3.20.101
2014 Feb  920:58:32.843484 pim: vPC: Inserted route (100.3.20.101/32, 225.131.38.2/32) (VRF default) to MRIB
delete-buffer
```

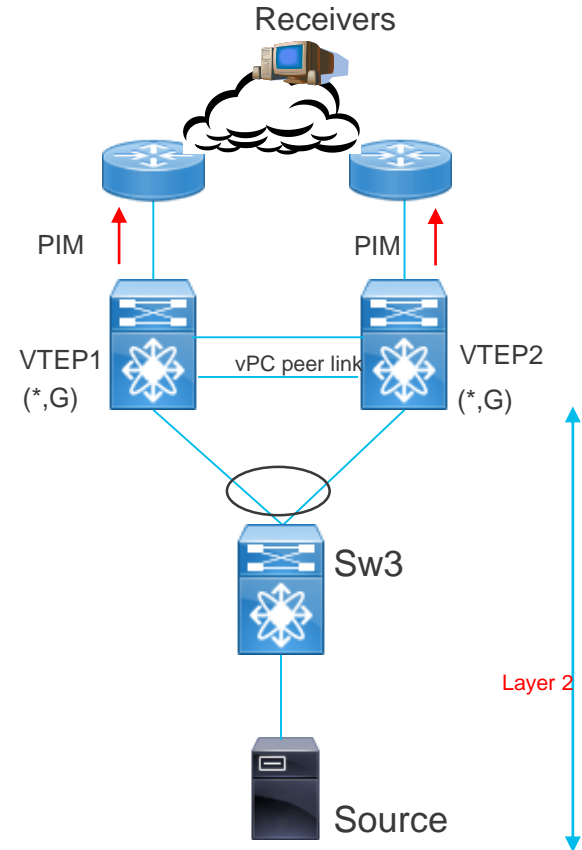
# Troubleshooting Multicast in vPC Environments

## Ingress vPC

### Problem Symptom:

What to do If receivers are not receiving the multicast from Source in vPC?

- Remote receivers.
- Source is connected to vPC Peers
- vPC vlan defined as any vlan that is configured on vPC peer-link



# Troubleshooting Multicast in vPC Environments

## Ingress vPC: SG Creation

!!!! “Ingress vPC” means ingress on vlan carried on vPC-PL !!!!

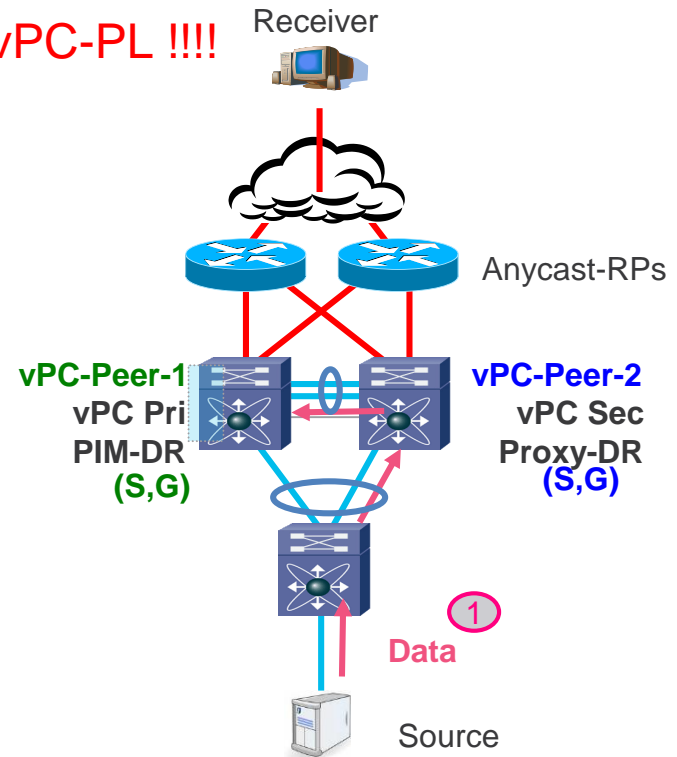
1. Source begins transmitting ingress vPC-Peer-2
  - Forwarded across peer link since it is an MRouter port
  - Both boxes have (S,G)
    - DR would register with RP here (standard procedure)

```
Nexus-1# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 00:00:14, ip pim
  Incoming interface: Vlan12, RPF nbr: 10.12.0.5
  Outgoing interface list: (count: 0)

Nexus-2# show ip mroute 239.12.0.5 10.12.0.5

(10.12.0.5/32, 239.12.0.5/32), uptime: 00:00:31, pim ip
  Incoming interface: Vlan12, RPF nbr: 10.12.0.5, internal
  Outgoing interface list: (count: 0)
```

Nothing new here, standard non-vPC steps



# Troubleshooting Multicast in vPC Environments

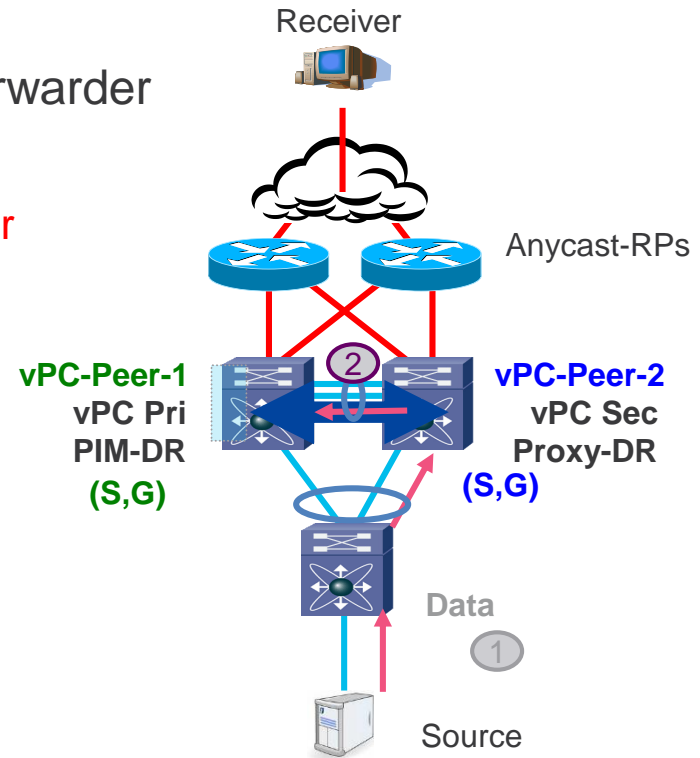
## Ingress vPC: Dual Forwarder

2. Upon creation of (S,G), vPC peers negotiate Forwarder
  - Both realize source is vPC-connected
  - Install forwarding entry as Win-Force / **Dual-Forwarder**

```
Nexus-1# show ip pim internal vpc rpf-source
Source: 10.12.0.5
  Pref/Metric: 0/0
  Source role: primary
  Forwarding state: Win-force (forwarding)

Nexus-2# show ip pim internal vpc rpf-source
Source: 10.12.0.5
  Pref/Metric: 0/0
  Source role: secondary
  Forwarding state: Win-force (forwarding)
```

So which device will then forward the stream?



# Troubleshooting Multicast in vPC Environments

## Ingress vPC: Egress L3

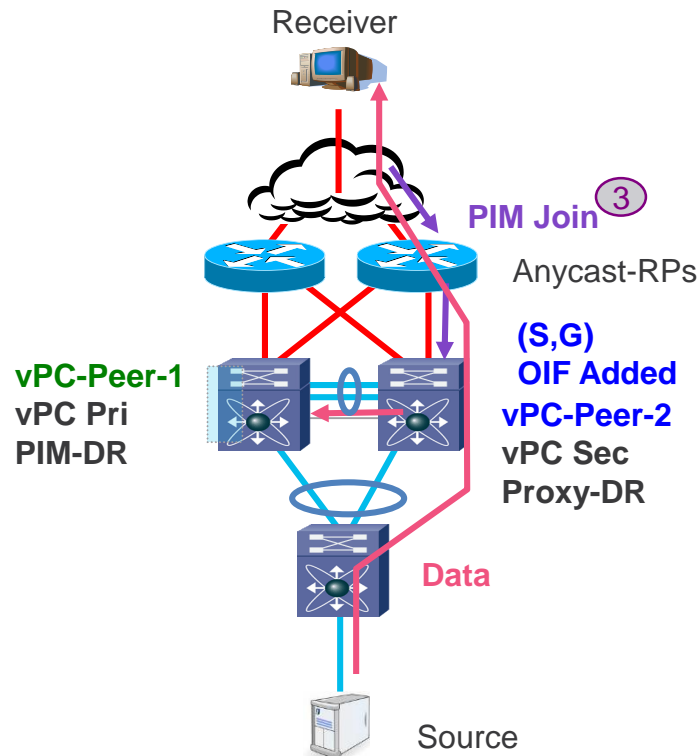
If L3 Receiver, no different then standard PIM-SM

### 3. PIM join sent from L3 cloud

- Only the peer that receives PIM Join installs OIF
  - L3 does not sync, only L2 info
- Traffic flows to receiver

```
Nexus-1# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 16:54:32, ip pim
Incoming interface: Vlan12, RPF nbr: 10.12.0.5
Outgoing interface list: (count: 0)

Nexus-2# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 16:56:09, pim ip
Incoming interface: Vlan12, RPF nbr: 10.12.0.5, internal
Outgoing interface list: (count: 1)
Ethernet7/37, uptime: 00:01:27, pim
```



# Troubleshooting Multicast in vPC Environments

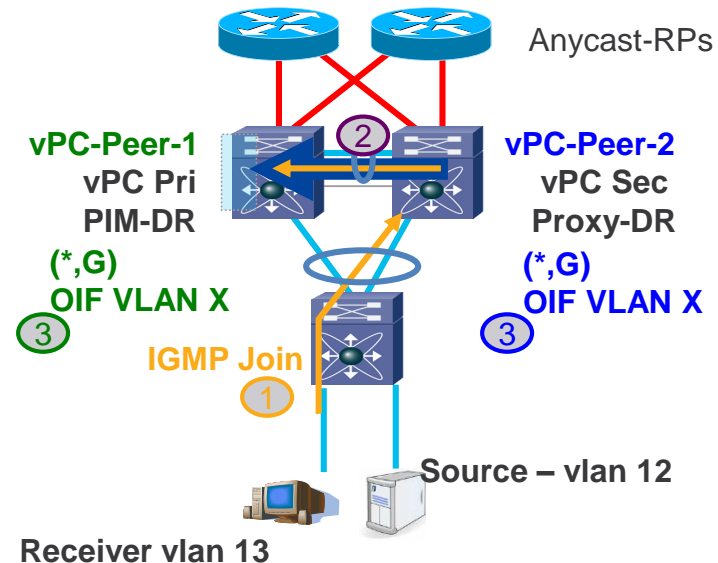
## Ingress vPC: Egress vPC

1. If VPC receiver IGMP Join enters Peer-2
2. Peer-2 encapsulates IGMP in CFS, sends to Peer-1
3. Both peers install OIF

```
Nexus-1# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 17:53:51, ip pim mrib
  Incoming interface: Vlan12, RPF nbr: 10.12.0.5
  Outgoing interface list: (count: 1)
    Vlan13, uptime: 00:00:14, mrib

Nexus-2# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 17:54:21, pim ip mrib
  Incoming interface: Vlan12, RPF nbr: 10.12.0.5, internal
  Outgoing interface list: (count: 1)
    Vlan13, uptime: 00:00:19, mrib
```

Here comes the curveball...





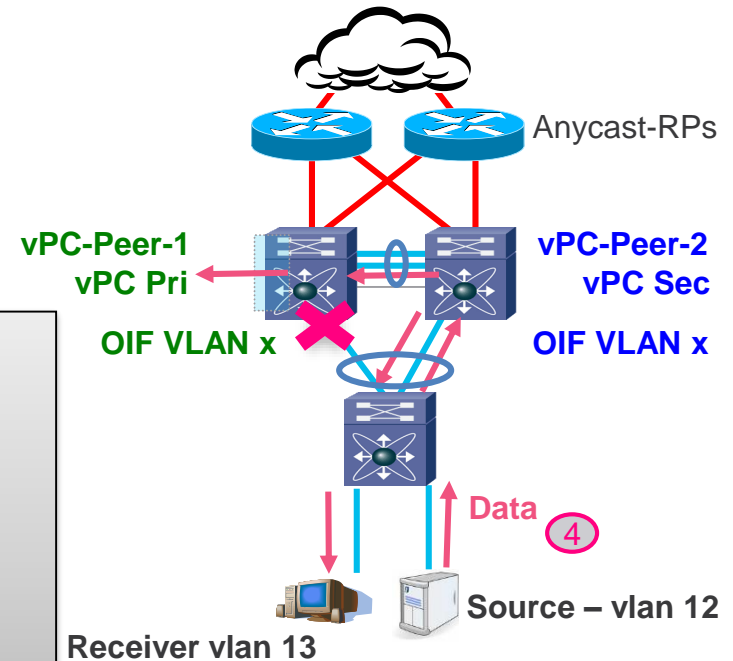
# Troubleshooting Multicast in vPC Environments

## Ingress vPC: Dual-Forwarder

4. Data packets enter Peer-2, creates (S,G)
  - a. Forwarded across peer link *'in original VLAN only'*
  - b. Replicated to OIF VLAN by Peer-2
  - c. Only tx **out local vPC / orphan ports** (*not peer-link*)
5. Data packets enter Peer-1 from peer-link (4a above)
  - a. Replicated to **OIF VLAN** – *'sent only to orphan ports'*
  - b. vPC-bit blocked on vPC port prevents dups

```
Nexus-1# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 17:53:51, ip pim mrrib
Incoming interface: Vlan12, RPF nbr: 10.12.0.5
Outgoing interface list: (count: 1)
  Vlan13, uptime: 00:00:14, mrrib

Nexus-2# show ip mroute 239.12.0.5 10.12.0.5
(10.12.0.5/32, 239.12.0.5/32), uptime: 17:54:21, pim ip mrrib
Incoming interface: Vlan12, RPF nbr: 10.12.0.5
Outgoing interface list: (count: 1)
  Vlan13, uptime: 00:00:19, mrrib
```



# Troubleshooting Multicast in vPC Environment

## vPC Command Cheat Sheet



- Standard Commands

- `show ip igmp snooping groups (x) | exc */*`
- `show ip igmp route (x)`
- `show ip igmp snooping internal event-history vpc`
- `show ip mroute detail`
- `show ip pim internal vpc rpf-source`
- `show ip pim event-history vpc`
- `show ip pim internal event-history join-prune`

- Tech Supports (capture from both vPC peers)

- `show tech-support vpc > bootflash:tech-vpc`
- `show tech-support ip multicast > bootflash:tech-ipmc`
- `show tech-support routing ip unicast detail > bootflash:tech-ucast`
- `show tech-support m2rib > bootflash:tech-m2rib`
- `show tech-support m2fib > bootflash:tech-m2fib`
- `show tech-support forwarding l2 multicast > bootflash:tech-l2mcast`
- `show tech-support forwarding l3 multicast > bootflash:tech-l3mcast`
- `show tech-support cfs > bootflash:tech-cfs`

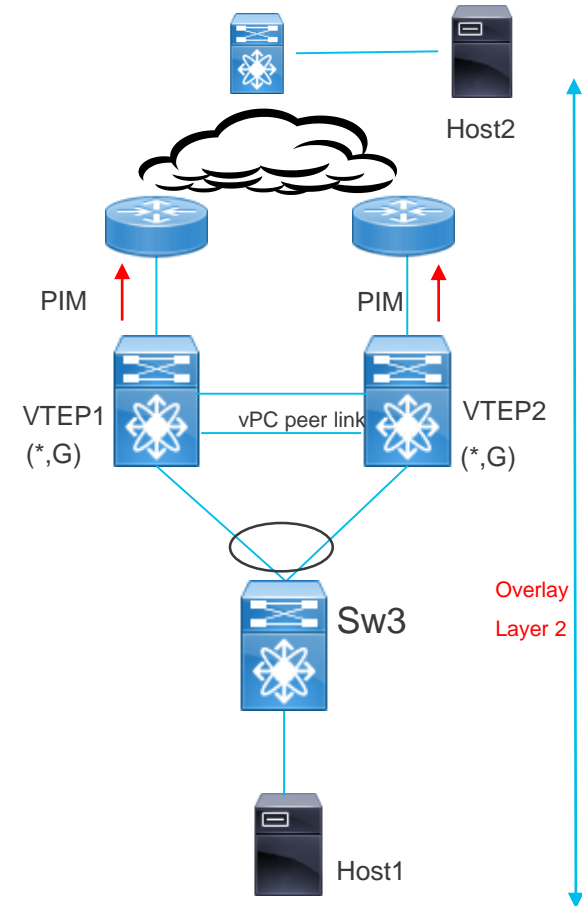
# NXOS Multicast Platform Independent Troubleshooting: VxLAN Environment

# Troubleshooting Multicast in VxLAN Environments

## Problem Symptom:

Overlay BUM traffic is not working between Host1 and Host2?

- Underlay Multicast between VTEPs
- Source and receivers are both VTEPs devices.

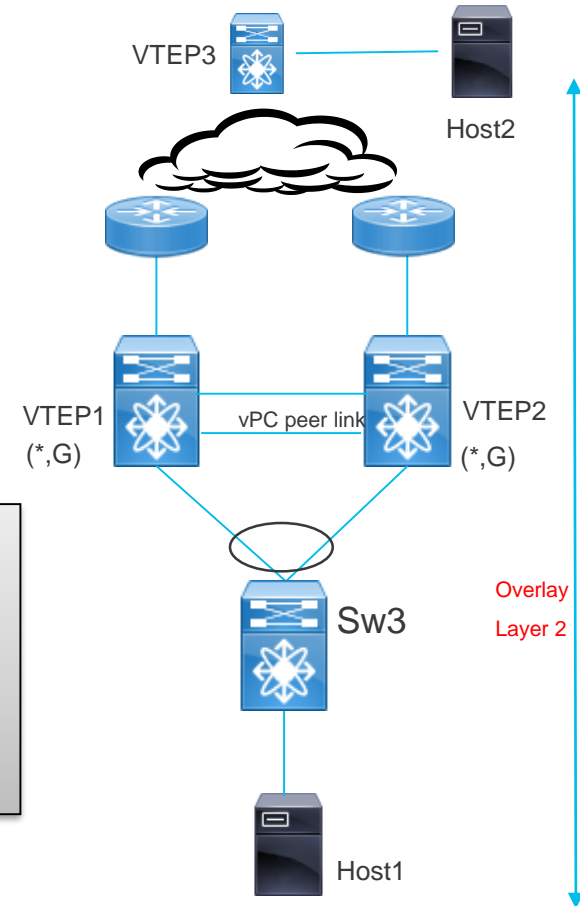


# Troubleshooting Multicast in VxLAN Environments

- Each VNI will be associated with a multicast group.
  - VLAN/VNI/Group Mapping
- Check if the VTEPs has (\*,G) entry for the VNI Multicast group.

```
VTEP2# show nve vni
Codes: CP - Control Plane          DP - Data Plane
      UC - Unconfigured           SA - Suppress ARP
      SU - Suppress Unknown Unicast
Interface VNI      Multicast-group  State Mode Type [BD/VRF]  Flags
-----
nve1     100100      239.1.1.1        Up   CP   L2 [100]
nve1     100200      239.2.2.2        Up   CP   L2 [200]
nve1     100300      n/a              Up   CP   L3 [TEST]
```

VNI-Mcast  
mapping



# Troubleshooting Multicast in VxLAN Environments

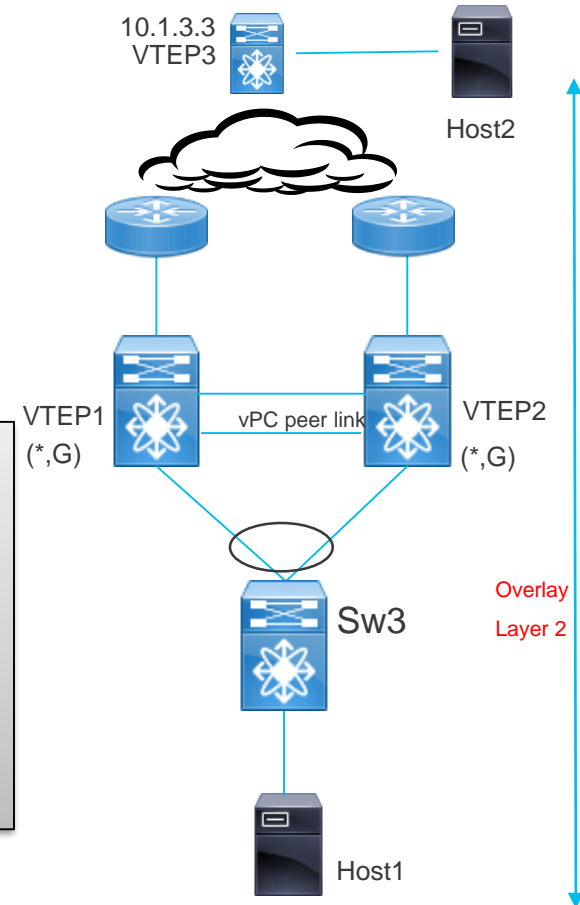
- Follow the traditional multicast troubleshooting.
- Each VTEP should have (\*,G) for the group and (S,G) for each remote VTEPs

```
VTEP2# show ip mroute 239.1.1.1
IP Multicast Routing Table for VRF "default"

(*, 239.1.1.1/32), uptime: 1d04h, nve ip pim
Incoming interface: Ethernet1/6, RPF nbr: 10.1.24.4
Outgoing interface list: (count: 1)
  nve1, uptime: 1d04h, nve

(10.1.3.3/32, 239.1.1.1/32), uptime: 23:20:42, ip mrib pim
Incoming interface: Ethernet1/6, RPF nbr: 10.1.24.4
Outgoing interface list: (count: 1)
  nve1, uptime: 23:20:42, mrib
```

(S,G) entry  
from remote  
VTEP



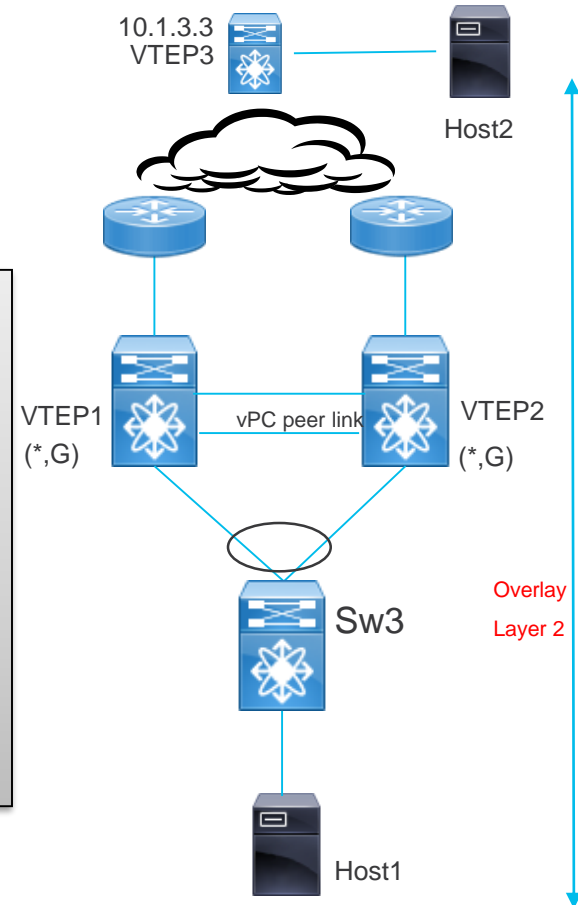
# Troubleshooting Multicast in VxLAN Environments

## Missing (S,G) in all VTEPs

- Ensure (S,G) is created on the originating VTEP.

```
VTEP3# show ip mroute 239.1.1.1 10.1.3.3 detail
<removed>
(10.1.3.3/32, 239.1.1.1/32) Route ptr: 0x704e7490 , uptime: 1d04h, nve(0)
mrib(0) ip(0) pim(1)
  RPF-Source: 10.1.3.3 [0/0]
  Data Created: No
  Received Register stop
  VXLAN Flags
    VXLAN Encap
  VPC Flags
    RPF-Source Forwarder
  Stats: 509860/48150846 [Packets/Bytes], 42.711 kbps
  Stats: Active Flow
  Incoming interface: loopback0, RPF nbr: 10.1.34.1
  Outgoing interface list: (count: 1) (bridge_only: 0)
    Ethernet1/6, uptime: 00:00:11, pim
```

VxLAN  
Encap



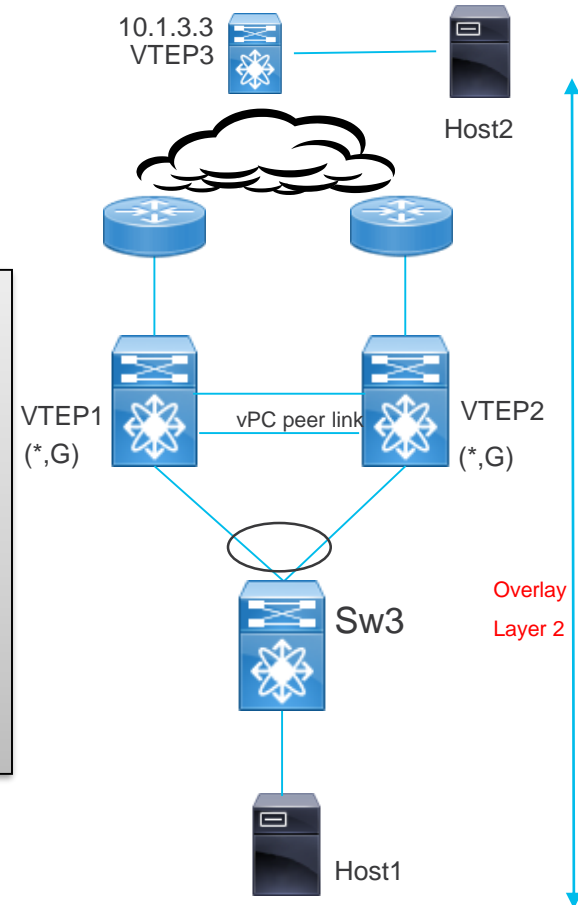
# Troubleshooting Multicast in VxLAN Environments

## (S,G) present in all VTEPs

- Ensure (S,G) is marked with VxLAN Decap in remote VTEPs.

```
VTEP2# show ip mroute 239.1.1.1 10.1.3.3 detail
<removed>
(10.1.3.3/32, 239.1.1.1/32) Route ptr: 0x704e7490 , uptime: 1d04h, nve(0)
mrib(0) ip(0) pim(1)
  RPF-Source: 10.1.54.4 [81/110]
  Data Created: No
VXLAN Flags
  VXLAN Decap
VPC Flags
  RPF-Source Forwarder
Stats: 509860/48150846 [Packets/Bytes], 42.711 kbps
Stats: Active Flow
Incoming interface: loopback0, RPF nbr: 10.1.34.1
Outgoing interface list: (count: 1) (bridge_only: 0)
  nve1, uptime: 00:00:11, mrib
```

VxLAN  
Decap





# Troubleshooting Multicast in VxLAN Environments

## Ethalyzer

- Ethalyzer is useful to verify the state entry creations.
- Utilize Ethalyzer to verify **CPU tx/rx** packets
- Use filters for better troubleshooting

```
VTEP# ethalyzer local interface inband display-filter "ip.dst==239.1.1.1&&udp.port==4789
2017-11-23 20:25:02.007844 10.1.3.3 -> 239.1.1.1  UDP Source port: 43124  Destination port: 4789
```

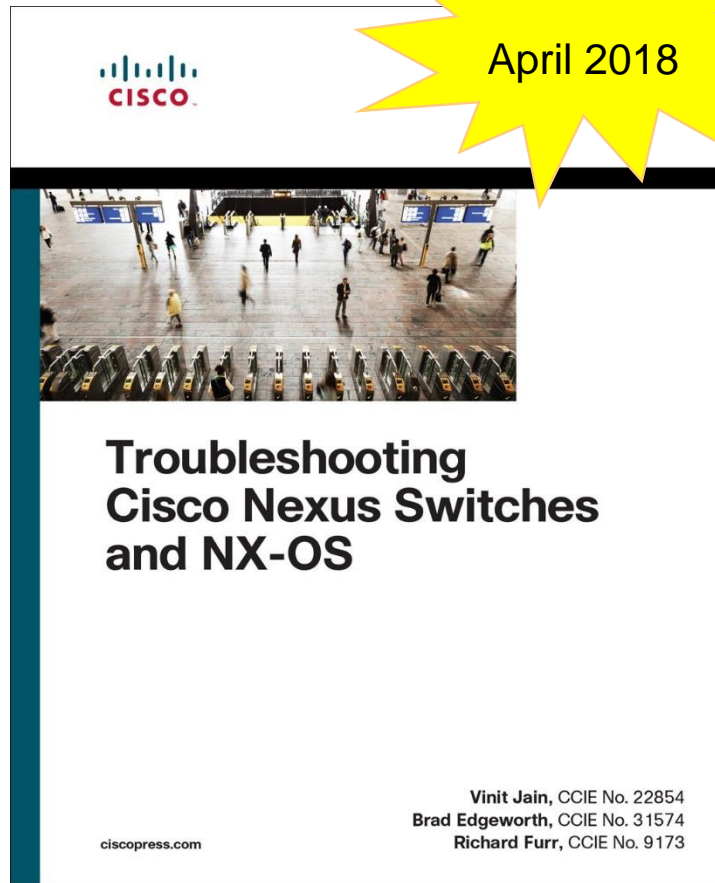
```
VTEP# ethalyzer local interface inband display-filter "ip.dst==239.1.1.1&&udp.port==4789 detail
```

- PIM Join / Prune Capture Example

```
Nexus# ethalyzer local interface inband capture-filter "src 10.1.3.3 and ip proto 103"
2017-11-22 11:29:34.046511 10.1.3.3 -> 224.0.0.13  PIMv2 Hello
2017-11-22 11:29:59.966210 10.1.3.3 -> 224.0.0.13  PIMv2 Join/Prune
```

## Recommended Reading

- Single source for troubleshooting problems on Nexus Switches
- Learn the techniques used by actual Cisco TAC Engineers.
- Covers the following topics: VLANs, PVLANS, STP, vPC, FabricPath, VXLAN, OTV, EIGRP, OSPF, IS-IS, Multicast, High Availability and Network Programability.
- Order here: <http://cs.co/TShootNexus>



# Cisco Spark

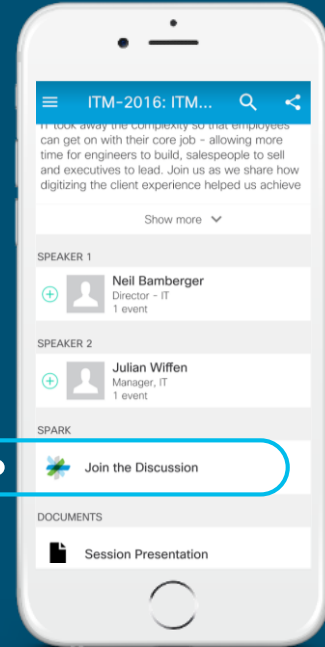


## Questions?

Use Cisco Spark to communicate with the speaker after the session

## How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space



[cs.co/cicolivebot#SESSION ID](https://cs.co/cicolivebot#SESSION ID)

- Please complete your Online Session Evaluations after each session
- Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt
- All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don't forget: Cisco Live sessions will be available for viewing on-demand after the event at [www.ciscolive.com/global/on-demand-library/](http://www.ciscolive.com/global/on-demand-library/).

## Complete Your Online Session Evaluation



# Continue Your Education

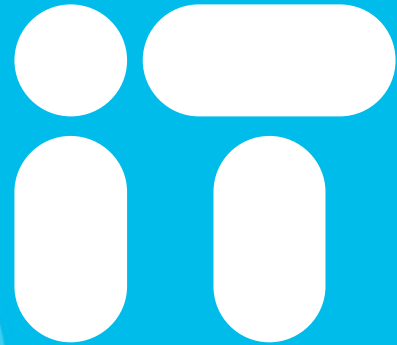
- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Tech Circle
- Meet the Engineer 1:1 meetings
- Related sessions



Thank you



You're



Cisco *live!*