EVPN VXLAN Interoperability Between NXOS and Junos OS

Author: Aninda Chatterjee

09-May-2023

Table of Contents

Executive Summary

This document is a technical exploration of interoperability issues between Cisco's NXOS and Juniper's Junos OS when running a BGP-based EVPN VXLAN fabric. The target audience of this document is technical practitioners who are considering or currently attempting to overcome challenges associated with creating a heterogeneous data center network fabric consisting of switches running Cisco's NXOS and switches running Juniper's Junos OS.

The goal of this document is to demonstrate how the two network operating systems can successfully interoperate. This document aims to provide an understanding of potential points of concern, as well as examples of how to overcome these issues.

Introduction

Organizations with more than one network vendor have always occurred, even in data center networks. Acquisitions and mergers, for example, regularly present interoperability considerations to IT teams. Recently, supply chain considerations and maintaining relationships with multiple vendors for critical infrastructure have begun to introduce more IT teams to heterogeneous networking environments.

The most common approach to multi-vendor networks is for individual fabrics to be restricted to a single vendor, communicating with other fabrics using Data Center Interconnect (DCI). Multi-vendor fabrics are less common but do happen. Cisco and Juniper are two of the largest network switch vendors in the world, and there are many organizations around the world that have deployed switches from both vendors.

It is important to understand how Cisco's NXOS and Juniper's Junos OS work together. Specifically, the details of EVPN VXLAN interoperability are relevant to data center operators, as a BGP-based EVPN-VXLAN solution is considered Juniper best practice for data center networks. This document will explore what works, what doesn't work, and more importantly the how and the why behind both.

Use Cases

The guide includes the following uses cases:

- A bridged overlay fabric with NXOS vPC and Junos OS ESI LAG. In this use case, we'll explore interoperability issues seen with NXOS as the spines and demonstrate a solution with relevant configuration. NXOS devices continue to be used as fabric spines throughout the document.
- Asymmetric IRB between NXOS and Junos OS.
- Symmetric IRB on NXOS and Asymmetric IRB on Junos OS.
- Symmetric IRB on NXOS and Symmetric IRB on Junos OS.
- DCI between a NXOS fabric and a Junos OS fabric for a bridged overlay.

This whitepaper describes the configuration for these use cases to work, provides configuration examples, as well as packet captures and packet walks.

Bridged Overlay with NXOS vPC and Junos OS ES-LAG

This use case consists of a data center network fabric with two Cisco NXOS switches forming a virtual Port Channels (vPC) pair, and two Junos OS devices as Ethernet Switch Identifier (ESI) Link Aggregation Group (LAG) peers. Both pairs of switches are operating as leaf switches.

Multiple hosts are connected to the leaf switches:

- h1 is connected via a vPC to leaf1 and leaf2 (the Cisco pair)
- h2 is single homed to leaf1 (one of the Cisco switches)
- h3 is connected to leaf3 and leaf4 (the Juniper pair) via ESI-LAG (using EVPN LAG multihoming)

Overview and Topology

For the bridged overlay use case, we will use the following topology:

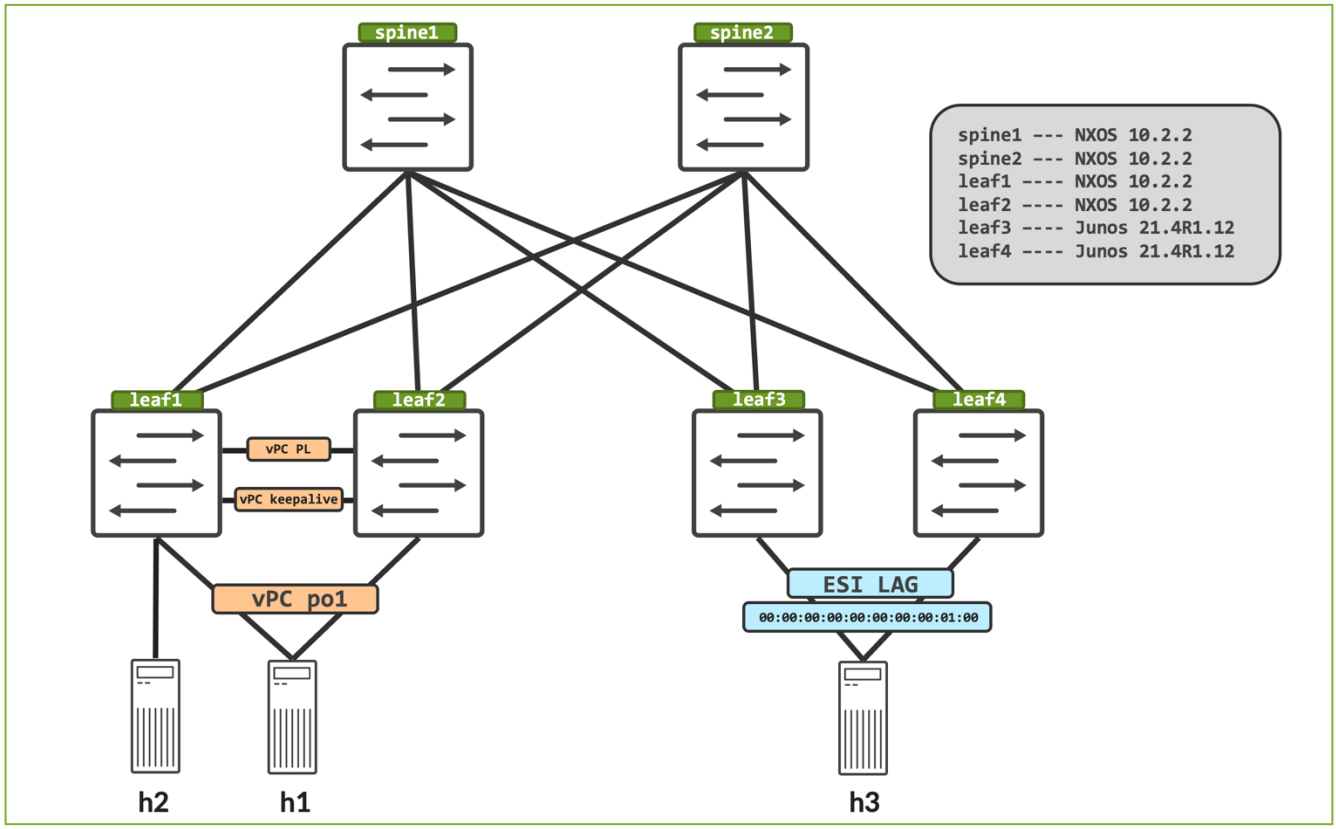


Figure 1: Topology Diagram for Bridged Overlay

Documentation IPv4 addresses (https://datatracker.ietf.org/doc/rfc5737/) are used to build this fabric as follows:

- The 192.0.2.0/24 range is used for loopbacks per fabric device.
- The 198.51.100.0/24 range is used for point-to-point /31 underlay links.
- The 203.0.113.0/24 range is used for host addressing, with the last octet being the respective host number (1 for h1, 2 for h2 and 3 for h3).

NXOS as Fabric Spines

Before we dive into the interoperability specifics between NXOS and Junos OS in a bridged overlay fabric, we need to consider the implications of using NXOS devices as spines. Considering our topology, let us assume we only have the two NXOS spines and the NXOS vPC pair connected. BGP peering is up for both the underlay and overlay, considering spine1 as an example here.

```
spine1# show bgp ipv4 unicast summary
BGP summary information for VRF default, address family IPv4 Unicast
BGP router identifier 192.0.2.11, local AS number 65500
BGP table version is 12, IPv4 Unicast config peers 4, capable peers 4
6 network entries and 7 paths using 2144 bytes of memory
BGP attribute entries [5/1760], BGP AS path entries [4/24]
BGP community entries [0/0], BGP clusterlist entries [0/0]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
198.51.100.0    4 65421    9455    9453       12    0    0    6d13h 2
198.51.100.2    4 65422    9455    9451       12    0    0    6d13h 2

*snip*

spine1# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 192.0.2.11, local AS number 65500
BGP table version is 6, L2VPN EVPN config peers 4, capable peers 2
0 network entries and 0 paths using 0 bytes of memory
BGP attribute entries [0/0], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [0/0]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.0.2.1       4 65421     580     578        6    0    0 09:31:11 0
192.0.2.2       4 65422     577     579        6    0    0 09:28:09 0

*snip*
```

Each leaf has an EVPN Type-3 IMET route in the BGP EVPN table, and it is sent to the spines:

```
leaf1# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 7, Local Router ID is 192.0.2.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.1:32867    (L2VNI 10100)
*>l[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                        100      32768 i


leaf2# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 7, Local Router ID is 192.0.2.2
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.2:32867    (L2VNI 10100)
*>l[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                        100      32768 i
```

However, we do not see it on the spines. The routes are rejected, with the following logs, confirming the same, from one of the spines.

```
*snip*

2022-12-04T09:47:54.823580000+00:00 [M 27] [bgp] E_DEBUG [bgp_af_process_nlri:7447] (default) PFX:
[L2VPN EVPN] Dropping prefix [3]:[0]:[32]:[192.0.2.12]/88 from peer 192.0.2.2, due to attribute policy
rejected
```

```
2022-12-04T09:47:31.046562000+00:00 [M 27] [bgp] E_DEBUG [bgp_af_process_nlri:7447] (default) PFX:
[L2VPN EVPN] Dropping prefix [3]:[0]:[32]:[192.0.2.12]/88 from peer 192.0.2.1, due to attribute policy
rejected

*snip*
```

The NXOS spines are rejecting the route because there is no matching import policy for the route-target attached to the BGP EVPN update -there is no L2VNI with a matching import route-target. This is expected since this is a lean spine that is used only for IP forwarding and transit – it does not have any VLANs/VNIs configured.

To allow these routes to be accepted, we need to configure the *'retain route-target all'* option under the L2VPN EVPN AFI/SAFI:

```
spine1(config)# router bgp 65500
spine1(config-router)# address-family l2vpn evpn
spine1(config-router-af)# retain route-target all

spine2(config)# router bgp 65500
spine2(config-router)# address-family l2vpn evpn
spine2(config-router-af)# retain route-target all
```

Note: If you want to retain only specific route-targets, this can also be done using a route-map instead of the 'all' keyword, where the route-map matches and permits the route-targets that need to be accepted.

After this is configured, the spines correctly accept these routes, with spine1 shown as an example below.

```
spine1# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 192.0.2.11, local AS number 65500
BGP table version is 8, L2VPN EVPN config peers 4, capable peers 2
2 network entries and 2 paths using 664 bytes of memory
BGP attribute entries [2/704], BGP AS path entries [2/12]
BGP community entries [0/0], BGP clusterlist entries [0/0]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.0.2.1       4 65421     584     581        8    0    0 09:32:55 1
192.0.2.2       4 65422     581     582        8    0    0 09:29:52 1

*snip*

spine1# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 8, Local Router ID is 192.0.2.11
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.1:32867
*>e[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                                        0 65421 i

Route Distinguisher: 192.0.2.2:32867
*>e[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                                        0 65422 i
```

A second interoperability issue is that, by default, Junos OS implements the VLAN-aware EVPN service type. This means that Junos OS based switches add a non-zero Ethernet Tag ID to the BGP EVPN NLRIs for specific EVPN route types, as seen in the packet capture below.

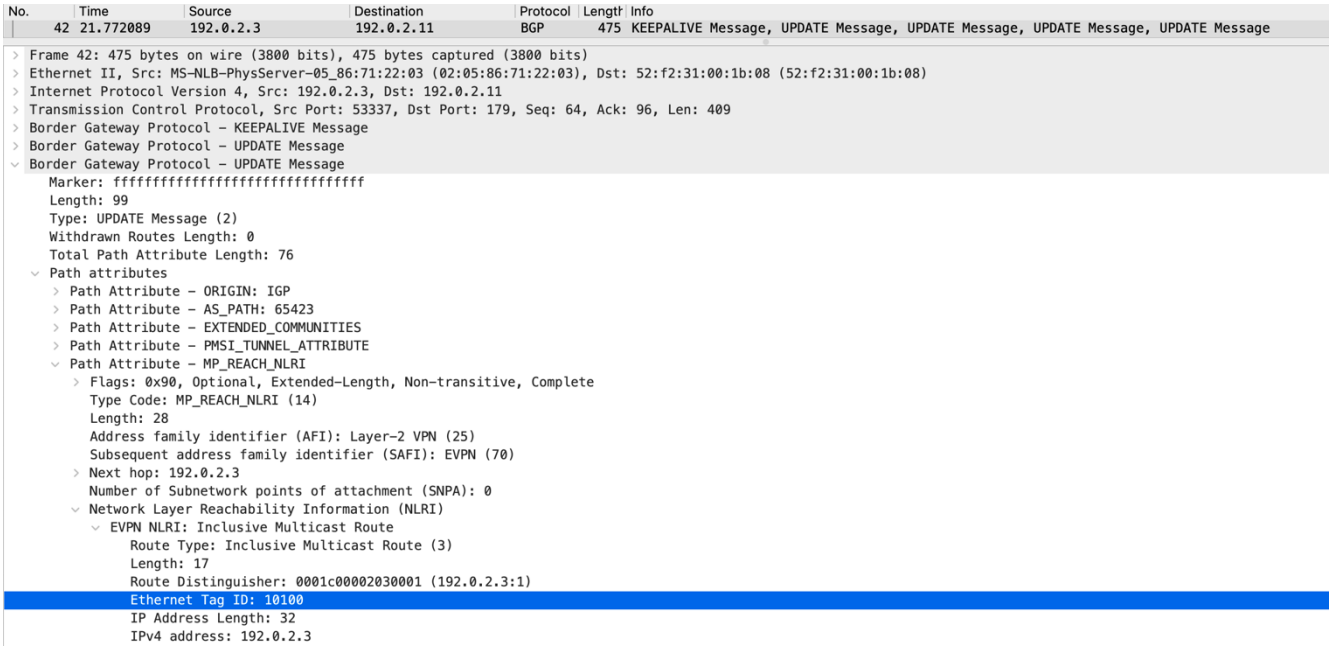


Figure 2: Packet Capture of a Non-Zero Ethernet Tag ID

NXOS does not know how to handle this, since it implements the VLAN-based EVPN service type. The NXOS default action is to drop any BGP EVPN NLRI, which has a non-zero Ethernet Tag ID:

```
*snip*

spine1# show bgp event-history events | grep 10100
2022 Dec 05 07:54:21.291679: E_DEBUG    bgp  [9756]: (default) UPD: [L2VPN EVPN] Received invalid
Ethernet Tag ID 10100 from peer 192.0.2.4, ignoring it
2022 Dec 05 07:53:55.100598: E_DEBUG    bgp  [9756]: (default) UPD: [L2VPN EVPN] Received invalid
Ethernet Tag ID 10100 from peer 192.0.2.3, ignoring it

*snip*
```

As the switches reject the BGP EVPN NLRIs, the routes will not propagate to leaf switches throughout the fabric. There are multiple ways to fix this. NXOS provides a configuration option to allow for such updates to be accepted. This configuration must be done on all NXOS switches, both spines and leafs.

Only changes to spine1 are shown in the following example:

```
spine1(config)# router bgp 65500
spine1(config-router)# address-family l2vpn evpn
spine1(config-router-af)# allow-vni-in-ethertag
```

With this configuration change made, leaf switches now correctly see the routes in their BGP RIB-In as well, because the spine switches are no longer dropping it. The NXOS leaf switches need the above configuration change, as well, in order to accept these routes.

```
leaf1# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 29, Local Router ID is 192.0.2.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
```

```
* e[1]:[0000.0000.0000.0000.0100]:[0x0]/152
                      192.0.2.4                                          0 65500 65424 i
*>e                   192.0.2.3                                          0 65500 65423 i
*>l[2]:[0]:[0]:[48]:[aac1.ab0c.8d71]:[0]:[0.0.0.0]/216
                      192.0.2.12                         100      32768 i
*>l[2]:[0]:[0]:[48]:[aac1.ab6d.65aa]:[0]:[0.0.0.0]/216
                      192.0.2.12                         100      32768 i
*>l[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                         100      32768 i
*>e[3]:[10100]:[32]:[192.0.2.3]/88
                      192.0.2.3                                          0 65500 65423 i
*>e[3]:[10100]:[32]:[192.0.2.4]/88
                      192.0.2.4                                          0 65500 65424 i

*snip*
```

For Junos OS based switches, the recommended way to accept these routes is to create a routing-instance of type mac-vrf and use the VLAN-based service type. This is a cleaner approach for interoperability. There are scale considerations associated with this design choice. There are only a limited number of routing-instances that can be created per platform.

A VLAN-based mac-vrf routing-instance can only map to one VLAN/BD. If additional VLANs are needed, you need to create their respective mac-vrf routing-instances as well.

As an example, on leaf3, this is the new routing-instance that we created:

```
admin@leaf3> show configuration routing-instances
v100_mac_vrf {
    instance-type mac-vrf;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list all;
        }
    }
    vtep-source-interface lo0.0;
    service-type vlan-based;
    interface ae0.0;
    route-distinguisher 192.0.2.3:1;
    vrf-target target:100:100;
    vlans {
        v100 {
            vlan-id 100;
            vxlan {
                vni 10100;
            }
        }
    }
}
```

Taking another packet capture now, we see the Ethernet Tag is set to 0, and the NXOS spines and leafs should accept this.

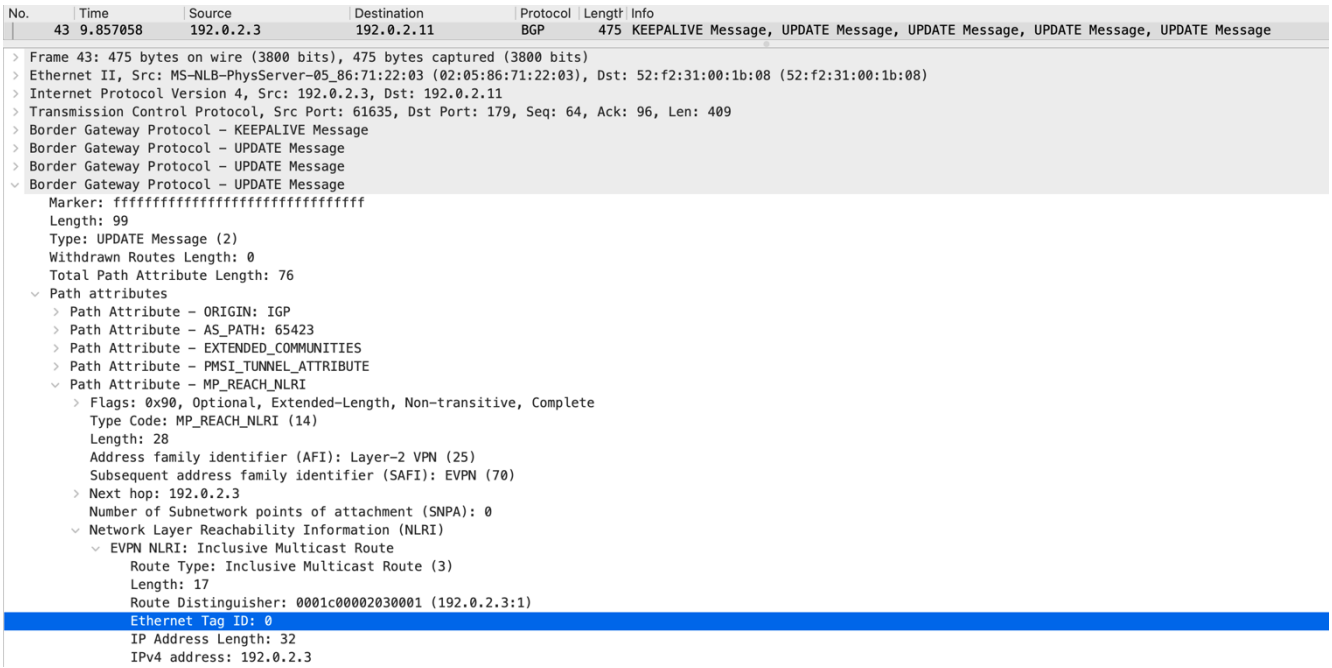


Figure 3: Packet Capture of a Zeroed Ethernet Tag ID

The EVPN Type-3 routes are now present on the leaf switches as well:

```
leaf1# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 69, Local Router ID is 192.0.2.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
* e[1]:[0000.0000.0000.0000.0100]:[0x0]/152
                      192.0.2.3                                          0 65500 65423 i
*>e                   192.0.2.4                                          0 65500 65424 i
*>e[2]:[0]:[0]:[48]:[aac1.ab50.3901]:[0]:[0.0.0.0]/216
                      192.0.2.3                                          0 65500 65423 i
*>l[2]:[0]:[0]:[48]:[aac1.ab86.2342]:[0]:[0.0.0.0]/216
                      192.0.2.12                             100      32768 i
*>l[2]:[0]:[0]:[48]:[aac1.abc5.e3c3]:[0]:[0.0.0.0]/216
                      192.0.2.12                             100      32768 i
*>e[3]:[0]:[32]:[192.0.2.3]/88
                      192.0.2.3                                          0 65500 65423 i
*>e[3]:[0]:[32]:[192.0.2.4]/88
                      192.0.2.4                                          0 65500 65424 i
*>l[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                             100      32768 i

*snip*
```

Interoperability Constraints for BUM Traffic

BUM traffic (Broadcast, Unknown Unicast and Multicast) is an acronym used as shorthand for multi-destination traffic. To support BUM traffic, you can use one of the following approaches:

- Deploy a multicast core (multicast in the underlay) to distribute such traffic towards the VTEPs (PEs) and eventually the endpoints.
- Use Ingress Replication, which packages the multi-destination traffic into unicast packets and sends one copy to every VTEP (PE).

Each approach has its own pros and cons. A multicast core reduces the load on the ingress leaf and distributes the responsibility of forwarding BUM traffic, but it adds complexity to the underlay and the need to maintain additional state.

Ingress Replication puts the burden of BUM traffic replication on the ingress leaf and increases traffic that is sent through the fabric core. With Ingress Replication, a copy of each multi-destination packet is created for each VTEP that had requested to be added to the flood list. There are further optimizations that are available and can be configured on Junos OS, such as Assisted Replication, Selective Multicast Ethernet (SMET) forwarding and Optimized Inter-Subnet Multicast (OISM) that reduce the flooding impact of ingress replication.

Each approach is different and since the mechanisms used to build state for each is different, the two cannot interoperate with each other. In NXOS, either a multicast group or ingress replication must be configured per VNI.

Let's consider the following configuration under leaf1:

```
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
```

VNI 10100 is defined under the NVE interface. We do not specify ingress replication via BGP as the mechanism for BUM traffic replication. This means that leaf1 no longer generates an EVPN Type-3 (IMET) route anymore.

```
leaf1# show bgp l2vpn evpn route-type 3
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
BGP routing table entry for [3]:[0]:[32]:[192.0.2.3]/88, version 151
Paths: (1 available, best #0)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is invalid(EVI down), no labeled nexthop
             Imported from 192.0.2.3:1:[3]:[0]:[32]:[192.0.2.3]/88
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:100:100 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 10100, Tunnel Id: 192.0.2.3

BGP routing table entry for [3]:[0]:[32]:[192.0.2.4]/88, version 152
Paths: (1 available, best #0)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is invalid(EVI down), no labeled nexthop
             Imported from 192.0.2.4:1:[3]:[0]:[32]:[192.0.2.4]/88
  AS-Path: 65500 65424 , path sourced external to AS
    192.0.2.4 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:100:100 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 10100, Tunnel Id: 192.0.2.4
```

```
Route Distinguisher: 192.0.2.3:1
BGP routing table entry for [3]:[0]:[32]:[192.0.2.3]/88, version 114
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.22 (192.0.2.22)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:100:100 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 10100, Tunnel Id: 192.0.2.3

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: L2-10100
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:100:100 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 10100, Tunnel Id: 192.0.2.3

  Path-id 1 not advertised to any peer

Route Distinguisher: 192.0.2.4:1
BGP routing table entry for [3]:[0]:[32]:[192.0.2.4]/88, version 121
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
  AS-Path: 65500 65424 , path sourced external to AS
    192.0.2.4 (metric 0) from 192.0.2.22 (192.0.2.22)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:100:100 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 10100, Tunnel Id: 192.0.2.4

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: L2-10100
  AS-Path: 65500 65424 , path sourced external to AS
    192.0.2.4 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:100:100 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 10100, Tunnel Id: 192.0.2.4

  Path-id 1 not advertised to any peer
```

As seen from the output above, the only Type-3 routes in the BGP EVPN table are the ones generated by leaf3 and leaf4. Remember that leaf3 and leaf4 rely on these Type-3 routes to build their ingress replication flood list. Lack of this route (from leaf1) means that leaf3/leaf4 do not create a remote VTEP entry for leaf1 and it is not in the flood list for replication:

```
admin@leaf3> show ethernet-switching vxlan-tunnel-end-point remote
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   192.0.2.3        lo0.0    0
 RVTEP-IP         L2-RTT                  IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 192.0.2.4        v100_mac_vrf            567       vtep.32770   1729    RNVE
    VNID          MC-Group-IP
    10100         0.0.0.0
```

This breaks the packet path for multi-destination traffic. Thus, to interoperate correctly with Junos OS, VNIs defined in NXOS must be configured for ingress replication as below:

```
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
```

Single BGP Session for Fabric Underlay/Overlay

A common misconception is that Junos OS cannot use a single BGP session for both the underlay and overlay of an EVPN VXLAN fabric. Junos OS does have the ability to do this.

Consider the following example from spine1 (NXOS) and leaf3 (Junos OS):

```
spine1# show run | sec "router bgp"
router bgp 65500
  router-id 192.0.2.11
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
  address-family l2vpn evpn
    retain route-target all
  neighbor 198.51.100.0
    remote-as 65421
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map nh-unchanged out
  neighbor 198.51.100.2
    remote-as 65422
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map nh-unchanged out
  neighbor 198.51.100.4
    remote-as 65423
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map nh-unchanged out
  neighbor 198.51.100.6
    remote-as 65424
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map nh-unchanged out
```

```
admin@leaf3> show configuration protocols bgp
group fabric {
    type external;
    export allow-loopback;
    peer-as 65500;
    multipath;
    neighbor 198.51.100.5 {
        family inet {
            unicast;
        }
        family evpn {
            signaling;
        }
```

```
    }
    neighbor 198.51.100.13 {
        family inet {
            unicast;
        }
        family evpn {
            signaling;
        }
    }
}
```

A BGP session is established for both inet (IPv4) and EVPN:

```
admin@leaf3> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                     10         10          0          0          0          0
bgp.evpn.0
                     20          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
198.51.100.5          65500        562        581       0       0     4:20:12 Establ
  inet.0: 5/5/5/0
  bgp.evpn.0: 0/10/0/0
  __default_evpn__.evpn.0: 0/1/0/0
  v100_mac_vrf.evpn.0: 0/9/0/0
198.51.100.13         65500        562        582       0       0     4:20:17 Establ
  inet.0: 5/5/5/0
  bgp.evpn.0: 0/10/0/0
  __default_evpn__.evpn.0: 0/1/0/0
  v100_mac_vrf.evpn.0: 0/9/0/0
```

You can confirm that this is over one TCP session only. You can see two sessions below, one for each spine that leaf3 is peering with.

```
admin@leaf3> show system connections | grep 179
tcp4       0      0  198.51.100.4.59837                     198.51.100.5.179
         ESTABLISHED
tcp4       0      0  198.51.100.12.50270                    198.51.100.13.179
          ESTABLISHED
tcp46      0      0  *.179                                  *.*
          LISTEN
tcp4       0      0  *.179                                  *.*
          LISTEN
```

The following is an example of a MAC learn and how it is distributed via BGP EVPN - leaf3 has learnt h3s MAC address:

```
admin@leaf3> show ethernet-switching table

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 1 entries, 1 learned
Routing instance : v100_mac_vrf
   Vlan                MAC                 MAC      Logical                SVLBNH/      Active
   name                address             flags    interface              VENH Index   source
   v100                aa:c1:ab:0c:1b:aa   DL       ae0.0
```

On leaf1, this is seen as an EVPN update and in the EVPN table.

```
leaf1# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 266, Local Router ID is 192.0.2.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
*>l[2]:[0]:[0]:[48]:[aac1.abe8.af2e]:[0]:[0.0.0.0]/216
                      192.0.2.12                          100      32768 i
*>l[2]:[0]:[0]:[48]:[aac1.abfa.6620]:[0]:[0.0.0.0]/216
                      192.0.2.12                          100      32768 i
*>l[3]:[0]:[32]:[192.0.2.12]/88
                      192.0.2.12                          100      32768 i

Route Distinguisher: 192.0.2.3:0
  e[1]:[0000.0000.0000.0000.0100]:[0xffffffff]/152
                      198.51.100.12                                    0 65500 65423 i
  e                   198.51.100.4                                     0 65500 65423 i
Route Distinguisher: 192.0.2.3:1
  e[1]:[0000.0000.0000.0000.0100]:[0x0]/152
                      198.51.100.4                                     0 65500 65423 i
  e                   198.51.100.12                                    0 65500 65423 i
  e[2]:[0]:[0]:[48]:[aac1.ab0c.1baa]:[0]:[0.0.0.0]/216
                      198.51.100.12                                    0 65500 65423 i
  e                   198.51.100.4                                     0 65500 65423 i
  e[3]:[0]:[32]:[192.0.2.3]/88
                      198.51.100.4                                     0 65500 65423 i
  e                   198.51.100.12                                    0 65500 65423 i

Route Distinguisher: 192.0.2.4:0
  e[1]:[0000.0000.0000.0000.0100]:[0xffffffff]/152
                      198.51.100.14                                    0 65500 65424 i
  e                   198.51.100.6                                     0 65500 65424 i

Route Distinguisher: 192.0.2.4:1
  e[1]:[0000.0000.0000.0000.0100]:[0x0]/152
                      198.51.100.14                                    0 65500 65424 i
  e                   198.51.100.6                                     0 65500 65424 i
  e[3]:[0]:[32]:[192.0.2.4]/88
                      198.51.100.14                                    0 65500 65424 i
  e                   198.51.100.6                                     0 65500 65424 i
```

Note that even though the routes are present, they are not selected as best. As a result, these routes are not considered for the forwarding table. Looking at the MAC route in detail, we can clearly see why.

```
leaf1# show bgp l2vpn evpn aa:c1:ab:0c:1b:aa
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.3:1
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.ab0c.1baa]:[0]:[0.0.0.0]/216, version 252
Paths: (2 available, best #0)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is invalid(rnh not resolved), no labeled nexthop
  AS-Path: 65500 65423 , path sourced external to AS
    198.51.100.12 (inaccessible, metric 4294967295) from 198.51.100.9 (192.0.2.22)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100

  Path type: external, path is invalid(rnh not resolved), no labeled nexthop
  AS-Path: 65500 65423 , path sourced external to AS
    198.51.100.4 (inaccessible, metric 4294967295) from 198.51.100.1 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100
```

Here, we can see that the output says, '*path is invalid*'. This is because the recursive next hop is not resolved resulting in the next hop not being accessible. The next hop should be leaf3s physical interface IP address facing the spines. This is the problem with a single session for the underlay/overlay. By default, Junos OS does not use the VTEP source as the next hop in EVPN NLRIs, while other vendors do this. Junos OS retains traditional BGP behavior, even for the EVPN family.

A policy must be used to change the next hop to the correct next hop for the EVPN family only. As an example, the following policy can be used:

```
admin@leaf3# show policy-options policy-statement evpn-nh
term set-nh {
    from protocol evpn;
    then {
        next-hop 192.0.2.3;
        accept;
    }
}
term reject {
    then reject;
}
```

To complete the configuration, the 'multihop' feature must be enabled under BGP as well, along with 'vpn-apply-export'. The reason that 'vpn-apply-export' is needed is because by default, when a VPN policy is applied, the routes are advertised using the VPN policy only, and any BGP policy is ignored.

It may not be obvious in the case of EVPN configuration, but the configuration for route-targets acts as a VPN policy. For the BGP policy to be processed as well, 'vpn-apply-export' is necessary. The final BGP configuration is:

```
admin@leaf3# show protocols bgp
group fabric {
    type external;
    multihop;
    export [ allow-loopback evpn-nh ];
    peer-as 65500;
    multipath;
    neighbor 198.51.100.5 {
        family inet {
            unicast;
        }
        family evpn {
            signaling;
        }
    }
    neighbor 198.51.100.13 {
        family inet {
            unicast;
        }
        family evpn {
            signaling;
        }
    }
    vpn-apply-export;
}
```

Now, on leaf1, the correct next hop is observed.

```
leaf1# show bgp l2vpn evpn aac1.ab0c.1baa
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.ab0c.1baa]:[0]:[0.0.0.0]/216, version 370
Paths: (2 available, best #2)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
             Imported from 192.0.2.3:1:[2]:[0]:[0]:[48]:[aac1.ab0c.1baa]:[0]:[0.0.0.0]/216
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 198.51.100.1 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.4:1:[2]:[0]:[0]:[48]:[aac1.ab0c.1baa]:[0]:[0.0.0.0]/216
  AS-Path: 65500 65424 , path sourced external to AS
```

```
    192.0.2.4 (metric 0) from 198.51.100.1 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100

*snip*
```

Deploying a Bridged Overlay Fabric with NXOS and Junos OS

With the assumption that the above interoperability issues are considered, and changes deployed to fix those, we can build our bridged overlay now.

The NXOS leaf switches (leaf1 and leaf2) are configured to be a vPC pair, with leaf1 being the primary. A dedicated VRF is created for the peer keepalive link (Ethernet1/10). Host h1 is connected via a vPC (port-channel1) to leaf1 and leaf2.

All configuration snippets shown are from one leaf only, with the expectation that the appropriate configuration is present on the other leaf as well.

```
vrf context vpc
  address-family ipv4 unicast
!
vpc domain 1
  peer-switch
  role priority 1
  peer-keepalive destination 198.51.100.17 source 198.51.100.16 vrf vpc
  peer-gateway
  ip arp synchronize
!
interface port-channel1
  switchport access vlan 100
  vpc 1
!
interface Ethernet1/3
  switchport access vlan 100
  channel-group 1 mode active
!
interface port-channel100
  switchport mode trunk
  spanning-tree port type network
  vpc peer-link
!
interface Ethernet1/10
  no switchport
  vrf member vpc
  ip address 198.51.100.16/31
  no shutdown
!
interface Ethernet1/11
  switchport mode trunk
  channel-group 100 mode active
!
```

BGP is used for both the underlay and overlay. The underlay peering uses the point-to-point addresses as neighbors while the overlay peering uses the loopbacks. It is important to note that as per Cisco vPC VXLAN requirements, the loopback on the vPC pair must share a common secondary IP address. This secondary IP address is used as the VTEP source/next-hop while advertising EVPN routes (including MAC/IP information for orphan devices as well).

The only exception to this is EVPN Type-5 routes where IP prefixes can be behind only one of the vPC switches in some designs, thus making it necessary to use the unique loopback IP address instead of the common secondary IP address.

```
interface loopback0
  ip address 192.0.2.1/32
```

```
  ip address 192.0.2.12/32 secondary
!
route-map allow-loopback permit 10
  match interface loopback0
!
router bgp 65421
  router-id 192.0.2.1
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
    maximum-paths 4
  address-family l2vpn evpn
    advertise-pip
  template peer evpn
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 192.0.2.11
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.22
    inherit peer evpn
    remote-as 65500
  neighbor 198.51.100.1
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.9
    remote-as 65500
    address-family ipv4 unicast
```

VLAN 100 is mapped to VNI 10100, and a Network Virtual Interface (nve1) is configured to allow VLAN 100 (VNI 10100) in ingress-replication mode, with BGP as the protocol to distribute the EVPN Type-3 IMET routes. VNI 10100 is also enabled for EVPN, configured with a route-distinguisher and export/import route-targets.

```
vlan 100
  vn-segment 10100
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
!
evpn
  vni 10100 l2
    rd 192.0.2.1:100
    route-target import 100:100
    route-target export 100:100
```

On the Junos OS devices, EVPN multihoming LAG is configured with a specific ESI. Each ESI LAG peer has a unique loopback.

```
admin@leaf3# show

*snip*

chassis {
    aggregated-devices {
        ethernet {
            device-count 5;
        }
    }
}
interfaces {

    *snip*

    ae0 {
```

```
        esi {
            00:00:00:00:00:00:00:00:01:00;
            all-active;
        }
        aggregated-ether-options {
            lacp {
                active;
                system-id 00:00:00:00:00:10;
            }
        }
        unit 0 {
            family ethernet-switching {
                interface-mode access;
                vlan {
                    members v100;
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.0.2.3/32;
            }
        }
    }
}
```

A routing-instance of type mac-vrf is created for VLAN 100:

```
routing-instances {
    v100_mac_vrf {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        interface ae0.0;
        route-distinguisher 192.0.2.3:1;
        vrf-target target:100:100;
        vlans {
            v100 {
                vlan-id 100;
                vxlan {
                    vni 10100;
                }
            }
        }
    }
}
```

Like the NXOS leafs, BGP is configured to use the point-to-point IPv4 addresses as neighbors for the underlay and the loopbacks for the overlay. The underlay and the overlay BGP configuration are separated into their own groups.

```
policy-options {
    policy-statement ECMP {
        then {
            load-balance per-flow;
        }
    }
    policy-statement allow-loopback {
        from interface lo0.0;
        then accept;
    }
}
routing-options {
    router-id 192.0.2.3;
```

```
    autonomous-system 65423;
    forwarding-table {
        export ECMP;
    }
}
protocols {
    bgp {
        group underlay {
            type external;
            family inet {
                unicast;
            }
            export allow-loopback;
            peer-as 65500;
            multipath;
            neighbor 198.51.100.5;
            neighbor 198.51.100.13;
        }
        group overlay {
            type external;
            multihop;
            local-address 192.0.2.3;
            family evpn {
                signaling;
            }
            peer-as 65500;
            neighbor 192.0.2.11;
            neighbor 192.0.2.22;
        }
    }

*snip*
```

Validating a Bridged Overlay Fabric with NXOS and Junos OS

The Junos OS leaf, leaf3, learns h3s MAC address over ae0 and installs it in the ethernet-switching table. This is inserted into the bgp.evpn.0 and v100_mac_vrf.evpn.0 table post learn and advertised to the NXOS spines:

```
admin@leaf3> show ethernet-switching table aa:c1:ab:b3:79:0c

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 3 entries, 3 learned
Routing instance : v100_mac_vrf
   Vlan                MAC                 MAC      Logical                 SVLBNH/      Active
   name                address             flags    interface               VENH Index   source
   v100                aa:c1:ab:b3:79:0c   DLR      ae0.0


admin@leaf3> show route table bgp.evpn.0 evpn-mac-address aa:c1:ab:b3:79:0c

bgp.evpn.0: 18 destinations, 30 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:192.0.2.3:1::0::aa:c1:ab:b3:79:0c/304 MAC/IP
                    *[EVPN/170] 00:44:13
                        Indirect
2:192.0.2.4:1::0::aa:c1:ab:b3:79:0c/304 MAC/IP
                    *[BGP/170] 00:44:14, localpref 100, from 192.0.2.11
                       AS path: 65500 65424 I, validation-state: unverified
                      >  to 198.51.100.5 via xe-0/0/0.0
                         to 198.51.100.13 via xe-0/0/1.0
                     [BGP/170] 00:44:14, localpref 100, from 192.0.2.22
                       AS path: 65500 65424 I, validation-state: unverified
                      >  to 198.51.100.5 via xe-0/0/0.0
                         to 198.51.100.13 via xe-0/0/1.0
2:192.0.2.3:1::0::aa:c1:ab:b3:79:0c::203.0.113.3/304 MAC/IP
                    *[EVPN/170] 00:43:50
                        Indirect
2:192.0.2.4:1::0::aa:c1:ab:b3:79:0c::203.0.113.3/304 MAC/IP
                    *[BGP/170] 00:01:45, localpref 100, from 192.0.2.11
```

```
                      AS path: 65500 65424 I, validation-state: unverified
                    >  to 198.51.100.5 via xe-0/0/0.0
                       to 198.51.100.13 via xe-0/0/1.0
                    [BGP/170] 00:01:45, localpref 100, from 192.0.2.22
                      AS path: 65500 65424 I, validation-state: unverified
                    >  to 198.51.100.5 via xe-0/0/0.0
                       to 198.51.100.13 via xe-0/0/1.0
```

This BGP EVPN update includes an Ethernet Segment ID as part of the NLRI itself and a route-target as an extended community. The next hop is the loopback of leaf3.

```
admin@leaf3> show route table bgp.evpn.0 evpn-mac-address aa:c1:ab:b3:79:0c extensive

bgp.evpn.0: 18 destinations, 30 routes (18 active, 0 holddown, 0 hidden)
2:192.0.2.3:1::0::aa:c1:ab:b3:79:0c/304 MAC/IP (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group overlay type External) Type 1 val 0xe9cd008 (adv_entry)
   Advertised metrics:
     Flags: Nexthop Change
     Nexthop: Self
     AS path: [65423] I
     Communities: target:100:100 encapsulation:vxlan(0x8)
    Advertise: 00000003
Path 2:192.0.2.3:1::0::aa:c1:ab:b3:79:0c
Vector len 4.  Val: 0
        *EVPN   Preference: 170
                Next hop type: Indirect, Next hop index: 0
                Address: 0xd22cb0c
                Next-hop reference count: 14
                Protocol next hop: 192.0.2.3
                Indirect next hop: 0x0 - INH Session ID: 0
                State: <Secondary Active Int Ext>
                Age: 44:52
                Validation State: unverified
                Task: v100_mac_vrf-evpn
                Announcement bits (1): 0-BGP_RT_Background
                AS path: I
                Communities: target:100:100 encapsulation:vxlan(0x8)
                Route Label: 10100
                ESI: 00:00:00:00:00:00:00:00:01:00
                Primary Routing Table: v100_mac_vrf.evpn.0
                Thread: junos-main

*snip*
```

The NXOS spines advertise these routes to the NXOS leafs. The NXOS leafs (with the correct import route-targets in place) can import these routes.

```
leaf1# show bgp l2vpn evpn aa:c1:ab:b3:79:0c
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.abb3.790c]:[0]:[0.0.0.0]/216, version 37
Paths: (2 available, best #2)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
             Imported from 192.0.2.3:1:[2]:[0]:[0]:[48]:[aac1.abb3.790c]:[0]:[0.0.0.0]/216
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.4:1:[2]:[0]:[0]:[48]:[aac1.abb3.790c]:[0]:[0.0.0.0]/216
  AS-Path: 65500 65424 , path sourced external to AS
    192.0.2.4 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
```

```
      ESI: 0000.0000.0000.0000.0100

  Path-id 1 not advertised to any peer
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.abb3.790c]:[32]:[203.0.113.3]/248, version 51
Paths: (2 available, best #2)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
             Imported from 192.0.2.3:1:[2]:[0]:[0]:[48]:[aac1.abb3.790c]:[32]:[203.0.113.3]/248
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.4:1:[2]:[0]:[0]:[48]:[aac1.abb3.790c]:[32]:[203.0.113.3]/248
  AS-Path: 65500 65424 , path sourced external to AS
    192.0.2.4 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0100

  Path-id 1 not advertised to any peer

*snip*
```

A successful BGP import pushes this to l2rib/l2route. Notice that for the MAC address in question (aac1.abb3.790c), the resolution type is ESI. This implies that the EVPN Type-1 route(s) must also be present and successfully imported for this to be resolved. The resolution results in a path list which can include multiple VTEPs (as seen below) since every ESI LAG peer should ideally advertise the EVPN Type-1 route for that Ethernet Segment.

```
leaf1# show l2route evpn mac evi vni 10100 detail

Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv (AD):Auto-Delete (D):Del Pending
(S):Stale (C):Clear, (Ps):Peer Sync (O):Re-Originated (Nho):NH-Override
(Asy):Asymmetric (Gw):Gateway
(Pf):Permanently-Frozen, (Orp): Orphan

(PipOrp): Directly connected Orphan to PIP based vPC BGW
(PipPeerOrp): Orphan connected to peer of PIP based vPC BGW
Topology    Mac Address    Prod   Flags         Seq No     Next-Hops
----------- -------------- ------ ------------- ---------- --------------------------------------
100         aac1.ab45.90e0 Local  L,            0          Eth1/4
            Route Resolution Type: Regular
            Forwarding State: Resolved
            Sent To: BGP
            SOO: 775043377

100         aac1.abb3.790c BGP    SplRcv        0          192.0.2.4 (Label: 10100)
            Route Resolution Type: ESI
            Forwarding State: Resolved (PL)
            Resultant PL: 192.0.2.3, 192.0.2.4
            Sent To: L2FM
            ESI : 0000.0000.0000.0000.0100
            Encap: 1

100         aac1.abfa.255e Local  L,            0          Po1
            Route Resolution Type: Regular
            Forwarding State: Resolved
            Sent To: BGP
            SOO: 775043377
```

This, in turn, is sent to L2FM which installs the MAC address.

```
leaf1# show mac address-table address aac1.abb3.790c
```

```
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
        (NA)- Not Applicable
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C  100     aac1.abb3.790c   dynamic  NA         F      F    nve1(192.0.2.3 192.0.2.4)
```

A similar control plane flow is expected for the hosts behind the NXOS vPC pair. There are different processes involved in Junos OS. The result is that the bgp.evpn.0 table on leaf3/leaf4 should have h2's MAC address.

```
admin@leaf3> show route table bgp.evpn.0 evpn-mac-address aac1.abfa.255e extensive

bgp.evpn.0: 18 destinations, 30 routes (18 active, 0 holddown, 0 hidden)
2:192.0.2.1:100::0::aa:c1:ab:fa:25:5e/304 MAC/IP (2 entries, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 192.0.2.1:100
                Next hop type: Indirect, Next hop index: 0
                Address: 0xd22cfb0
                Next-hop reference count: 24
                Source: 192.0.2.11
                Protocol next hop: 192.0.2.12
                Indirect next hop: 0x2 no-forward INH Session ID: 0
                State: <Active Ext>
                Local AS: 65423 Peer AS: 65500
                Age: 40:59      Metric2: 0
                Validation State: unverified
                Task: BGP_65500.192.0.2.11
                AS path: 65500 65421 I
                Communities: target:100:100 origin:192.0.2.12:0 encapsulation:vxlan(0x8)
                Import Accepted
                Route Label: 10100
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 192.0.2.11
                Secondary Tables: v100_mac_vrf.evpn.0
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 192.0.2.12
                        Indirect next hop: 0x2 no-forward INH Session ID: 0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 198.51.100.5 via xe-0/0/0.0
                                Session Id: 0
                                Next hop: 198.51.100.13 via xe-0/0/1.0
                                Session Id: 0
                                192.0.2.12/32 Originating RIB: inet.0
                                  Node path count: 1
                                  Forwarding nexthops: 2
                                        Next hop type: Router
                                        Next hop: 198.51.100.5 via xe-0/0/0.0
                                        Session Id: 0
                                        Next hop: 198.51.100.13 via xe-0/0/1.0
                                        Session Id: 0
         BGP    Preference: 170/-101
                Route Distinguisher: 192.0.2.1:100
                Next hop type: Indirect, Next hop index: 0
                Address: 0xd22cfb0
                Next-hop reference count: 24
                Source: 192.0.2.22
                Protocol next hop: 192.0.2.12
                Indirect next hop: 0x2 no-forward INH Session ID: 0
                State: <NotBest Ext Changed>
                Inactive reason: Not Best in its group - Active preferred
                Local AS: 65423 Peer AS: 65500
                Age: 40:59      Metric2: 0
                Validation State: unverified
                Task: BGP_65500.192.0.2.22
                AS path: 65500 65421 I
                Communities: target:100:100 origin:192.0.2.12:0 encapsulation:vxlan(0x8)
                Import Accepted
                Route Label: 10100
                ESI: 00:00:00:00:00:00:00:00:00:00
```

```
                Localpref: 100
                Router ID: 192.0.2.22
                Secondary Tables: v100_mac_vrf.evpn.0
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 192.0.2.12
                        Indirect next hop: 0x2 no-forward INH Session ID: 0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 198.51.100.5 via xe-0/0/0.0
                                Session Id: 0
                                Next hop: 198.51.100.13 via xe-0/0/1.0
                                Session Id: 0
                                192.0.2.12/32 Originating RIB: inet.0
                                  Node path count: 1
                                  Forwarding nexthops: 2
                                        Next hop type: Router
                                        Next hop: 198.51.100.5 via xe-0/0/0.0
                                        Session Id: 0
                                        Next hop: 198.51.100.13 via xe-0/0/1.0
                                        Session Id: 0

2:192.0.2.2:100::0::aa:c1:ab:fa:25:5e/304 MAC/IP (2 entries, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 192.0.2.2:100
                Next hop type: Indirect, Next hop index: 0
                Address: 0xd22cfb0
                Next-hop reference count: 24
                Source: 192.0.2.22
                Protocol next hop: 192.0.2.12
                Indirect next hop: 0x2 no-forward INH Session ID: 0
                State: <Active Ext>
                Local AS: 65423 Peer AS: 65500
                Age: 40:59      Metric2: 0
                Validation State: unverified
                Task: BGP_65500.192.0.2.22
                AS path: 65500 65422 I
                Communities: target:100:100 origin:192.0.2.12:0 encapsulation:vxlan(0x8)
                Import Accepted
                Route Label: 10100
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 192.0.2.22
                Secondary Tables: v100_mac_vrf.evpn.0
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 192.0.2.12
                        Indirect next hop: 0x2 no-forward INH Session ID: 0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 198.51.100.5 via xe-0/0/0.0
                                Session Id: 0
                                Next hop: 198.51.100.13 via xe-0/0/1.0
                                Session Id: 0
                                192.0.2.12/32 Originating RIB: inet.0
                                  Node path count: 1
                                  Forwarding nexthops: 2
                                        Next hop type: Router
                                        Next hop: 198.51.100.5 via xe-0/0/0.0
                                        Session Id: 0
                                        Next hop: 198.51.100.13 via xe-0/0/1.0
                                        Session Id: 0
         BGP    Preference: 170/-101
                Route Distinguisher: 192.0.2.2:100
                Next hop type: Indirect, Next hop index: 0
                Address: 0xd22cfb0
                Next-hop reference count: 24
                Source: 192.0.2.11
                Protocol next hop: 192.0.2.12
                Indirect next hop: 0x2 no-forward INH Session ID: 0
                State: <NotBest Ext Changed>
                Inactive reason: Not Best in its group - Active preferred
                Local AS: 65423 Peer AS: 65500
                Age: 40:59      Metric2: 0
                Validation State: unverified
```

```
                Task: BGP_65500.192.0.2.11
                AS path: 65500 65422 I
                Communities: target:100:100 origin:192.0.2.12:0 encapsulation:vxlan(0x8)
                Import Accepted
                Route Label: 10100
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 192.0.2.11
                Secondary Tables: v100_mac_vrf.evpn.0
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 192.0.2.12
                        Indirect next hop: 0x2 no-forward INH Session ID: 0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 198.51.100.5 via xe-0/0/0.0
                                Session Id: 0
                                Next hop: 198.51.100.13 via xe-0/0/1.0
                                Session Id: 0
                                192.0.2.12/32 Originating RIB: inet.0
                                  Node path count: 1
                                  Forwarding nexthops: 2
                                        Next hop type: Router
                                        Next hop: 198.51.100.5 via xe-0/0/0.0
                                        Session Id: 0
                                        Next hop: 198.51.100.13 via xe-0/0/1.0
                                        Session Id: 0
```

From the bgp.evpn.table, this gets pushed into the table specific to the routing-instance and eventually the ethernet-switching table.

```
admin@leaf3> show route table v100_mac_vrf.evpn.0 evpn-mac-address aac1.abfa.255e

v100_mac_vrf.evpn.0: 15 destinations, 26 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:192.0.2.1:100::0::aa:c1:ab:fa:25:5e/304 MAC/IP
                   *[BGP/170] 00:42:14, localpref 100, from 192.0.2.11
                      AS path: 65500 65421 I, validation-state: unverified
                    >  to 198.51.100.5 via xe-0/0/0.0
                       to 198.51.100.13 via xe-0/0/1.0
                    [BGP/170] 00:42:14, localpref 100, from 192.0.2.22
                      AS path: 65500 65421 I, validation-state: unverified
                    >  to 198.51.100.5 via xe-0/0/0.0
                       to 198.51.100.13 via xe-0/0/1.0
2:192.0.2.2:100::0::aa:c1:ab:fa:25:5e/304 MAC/IP
                   *[BGP/170] 00:42:14, localpref 100, from 192.0.2.22
                      AS path: 65500 65422 I, validation-state: unverified
                    >  to 198.51.100.5 via xe-0/0/0.0
                       to 198.51.100.13 via xe-0/0/1.0
                    [BGP/170] 00:42:14, localpref 100, from 192.0.2.11
                      AS path: 65500 65422 I, validation-state: unverified
                    >  to 198.51.100.5 via xe-0/0/0.0
                       to 198.51.100.13 via xe-0/0/1.0

admin@leaf3> show ethernet-switching table aac1.abfa.255e

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 3 entries, 3 learned
Routing instance : v100_mac_vrf
   Vlan                MAC                 MAC      Logical                 SVLBNH/      Active
   name                address             flags    interface               VENH Index   source
   v100                aa:c1:ab:fa:25:5e   DR       vtep.32769                           192.0.2.12
```

At the end of this, all hosts should now be able to ping each other:

```
root@h3:~# ping 203.0.113.1
PING 203.0.113.1 (203.0.113.1) 56(84) bytes of data.
64 bytes from 203.0.113.1: icmp_seq=1 ttl=64 time=19.1 ms
```

```
64 bytes from 203.0.113.1: icmp_seq=2 ttl=64 time=19.5 ms
64 bytes from 203.0.113.1: icmp_seq=3 ttl=64 time=17.3 ms
64 bytes from 203.0.113.1: icmp_seq=4 ttl=64 time=23.0 ms
^C
--- 203.0.113.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 17.327/19.737/23.026/2.072 ms
root@h3:~# ping 203.0.113.2
PING 203.0.113.2 (203.0.113.2) 56(84) bytes of data.
64 bytes from 203.0.113.2: icmp_seq=1 ttl=64 time=108 ms
64 bytes from 203.0.113.2: icmp_seq=2 ttl=64 time=157 ms
64 bytes from 203.0.113.2: icmp_seq=3 ttl=64 time=112 ms
64 bytes from 203.0.113.2: icmp_seq=4 ttl=64 time=179 ms
^C
--- 203.0.113.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 8ms
rtt min/avg/max/mdev = 108.076/139.012/179.466/30.244 ms
```

Asymmetric IRB on NXOS and Asymmetric IRB on Junos OS

Overview and Topology

This document expects the reader to be familiar with different approaches to Integrated Routing and Bridging (IRB) in an EVPN VXLAN fabric. As such, this document will not go into the finer details of Asymmetric and Symmetric IRB. The focus of this document is purely on the interoperability between the two network operating systems and IRB.

NXOS does not support Asymmetric IRB for EVPN VXLAN. Because of this, it is not possible to create a fabric where both NXOS and Junos OS are configured for Asymmetric IRB. However, it is important to understand why this fails.

Consider the following topology for this, and focus on communication between h1 and h4:

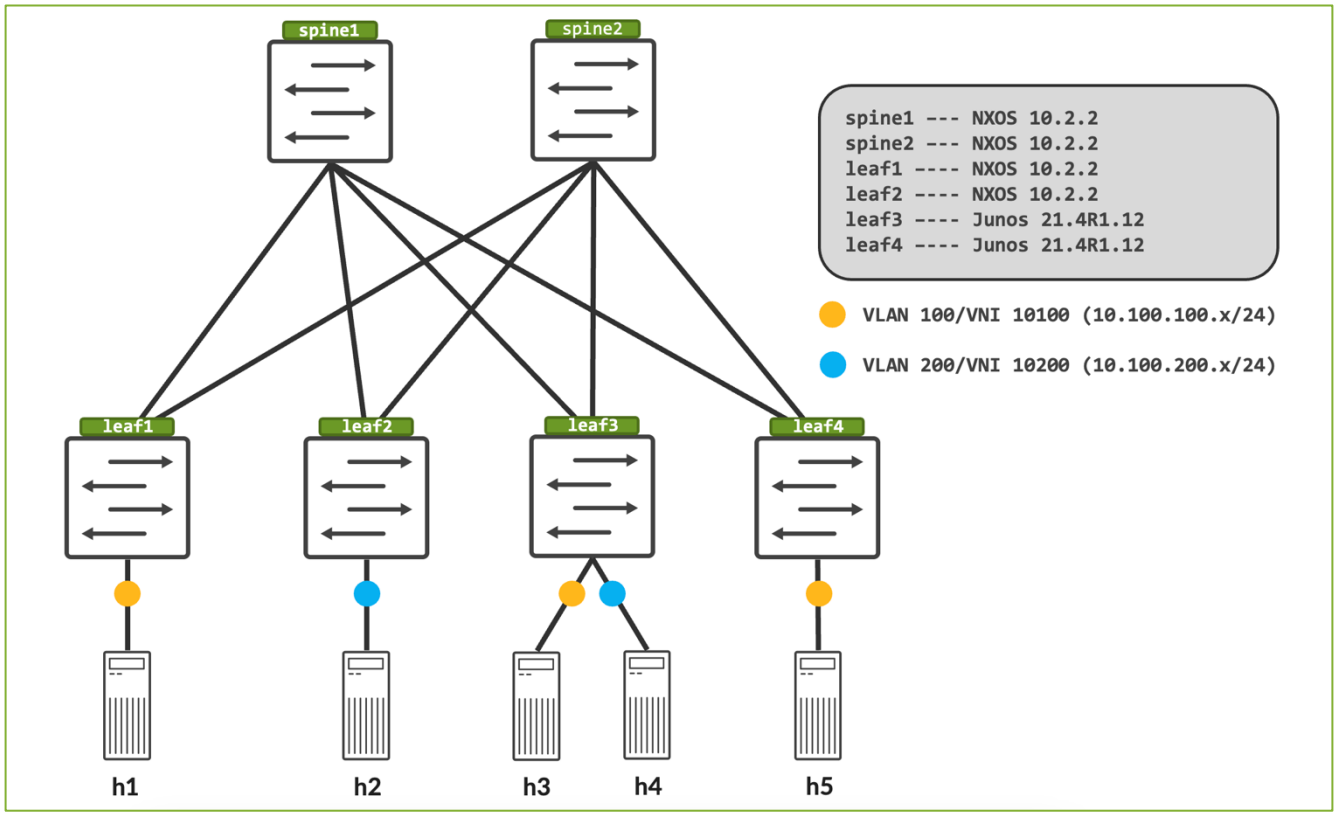


Figure 4: The Topology of the Fabric Used in this Document, with Assigned VNIs

Configuring Asymmetric IRB on NXOS

We'll configure leaf1 and leaf3 for Asymmetric IRB - this implies that all VLANs, VNIs and IRB interfaces exist on both leaf switches. In our case, this means VLAN 100, VNI 10100, VLAN 200, VNI 10200 and its respective IRB interfaces are configured on both leaf1 and leaf3.

For leaf1, which is running NXOS, the relevant configuration is as follows:

VLANs 100 and 200 are created and mapped to their respective VNIs. An anycast gateway MAC is also configured.

```
nv overlay evpn
feature bgp
feature fabric forwarding
feature interface-vlan
feature vn-segment-vlan-based
feature nv overlay
!
fabric forwarding anycast-gateway-mac 000a.000a.000a
vlan 1,100,200,300
vlan 100
  vn-segment 10100
vlan 200
  vn-segment 10200
!
```

IRB interfaces for these VLANs are created with the fabric forwarding mode as 'anycast-gateway' implying that the same IP address exists across all leafs for the IRB interface.

```
route-map allow-loopback permit 10
  match interface loopback0
!
interface Vlan100
  no shutdown
  ip address 10.100.100.254/24
  fabric forwarding mode anycast-gateway
!
interface Vlan200
  no shutdown
  ip address 10.100.200.254/24
  fabric forwarding mode anycast-gateway
!
```

Finally, an NVE interface is configured and these VNIs mapped under it, with ingress-replication. The same VNIs are also declared for EVPN, with their respective route-distinguishers and import/export route-targets configured.

```
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
  member vni 10200
    ingress-replication protocol bgp
!
interface loopback0
  ip address 192.0.2.1/32
!
router bgp 65421
  router-id 192.0.2.1
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
    maximum-paths 4
  address-family l2vpn evpn
    advertise-pip
  template peer evpn
```

```
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 192.0.2.11
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.22
    inherit peer evpn
    remote-as 65500
  neighbor 198.51.100.1
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.9
    remote-as 65500
    address-family ipv4 unicast
evpn
  vni 10100 l2
    rd 192.0.2.1:100
    route-target import 100:100
    route-target export 100:100
  vni 10200 l2
    rd 192.0.2.1:200
    route-target import 200:200
    route-target export 200:200
```

Configuring Asymmetric IRB on Junos OS

On leaf3, which is running Junos OS, we create the IRB interfaces for the respective VLANs with a unique physical address, and the same anycast gateway virtual address.

```
admin@leaf3# show

interfaces {
    irb {
        unit 100 {
            virtual-gateway-accept-data;
            family inet {
                address 10.100.100.252/24 {
                    virtual-gateway-address 10.100.100.254;
                }
            }
            virtual-gateway-v4-mac 00:0a:00:0a:00:0a;
        }
        unit 200 {
            virtual-gateway-accept-data;
            family inet {
                address 10.100.200.252/24 {
                    virtual-gateway-address 10.100.200.254;
                }
            }
            virtual-gateway-v4-mac 00:0a:00:0a:00:0a;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.0.2.3/32;
            }
        }
    }
}
```

For routing-instances of type mac-vrf create one routing-instance per VLAN. This is because we're enabling it for VLAN-based service type. Within each routing-instance, we configure the respective EVPN parameters, configure the VTEP source, and map the L2 VLANs to their IRB interfaces (along with their respective VNIs).

```
routing-instances {
    v100_mac_vrf {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway do-not-advertise;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        interface xe-0/0/2.0;
        route-distinguisher 192.0.2.3:100;
        vrf-target target:100:100;
        vlans {
            v100 {
                vlan-id 100;
                l3-interface irb.100;
                vxlan {
                    vni 10100;
                }
            }
        }
    }
    v200_mac_vrf {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway do-not-advertise;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        interface xe-0/0/3.0;
        route-distinguisher 192.0.2.3:200;
        vrf-target target:200:200;
        vlans {
            v200 {
                vlan-id 200;
                l3-interface irb.200;
                vxlan {
                    vni 10200;
                }
            }
        }
    }
}
```

Finally, BGP is configured as two groups: one for the underlay and another for the overlay. A policy is configured and applied (as an export policy) to the underlay group to advertise the loopback into BGP.

```
policy-options {
    policy-statement ECMP {
        then {
            load-balance per-flow;
        }
    }
    policy-statement allow-loopback {
        from interface lo0.0;
        then accept;
    }
}
routing-options {
    router-id 192.0.2.3;
    autonomous-system 65423;
    forwarding-table {
```

```
        export ECMP;
    }
}
protocols {
    bgp {
        group underlay {
            type external;
            family inet {
                unicast;
            }
            export allow-loopback;
            peer-as 65500;
            multipath;
            neighbor 198.51.100.5;
            neighbor 198.51.100.13;
        }
        group overlay {
            type external;
            multihop;
            local-address 192.0.2.3;
            family evpn {
                signaling;
            }
            peer-as 65500;
            neighbor 192.0.2.11;
            neighbor 192.0.2.22;
        }
    }
}
```

Since each leaf is configured with all IRBs, every VLAN is essentially directly connected to each leaf. This implies that both leaf1 and leaf3 can directly ARP for any host on VLAN100 or VLAN200.

Understanding Why Asymmetric IRB Does Not Work on NXOS

Host h1 has 10.100.100.254 as its default gateway. When pinging h4, it knows that the destination is in a different subnet, and it needs to send the packet to its default gateway. The ARP process provides the IP-MAC binding for 10.100.100.254 and h1 sends the ICMP request to leaf1.

On leaf1 (the NXOS leaf), since the destination MAC address is owned by it, a route lookup is done for 10.100.200.4. This hits the directly connected subnet route.

```
leaf1# show ip route 10.100.200.4
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.100.200.0/24, ubest/mbest: 1/0, attached
    *via 10.100.200.254, Vlan200, [0/0], 00:29:41, direct
```

leaf1 can now ARP for the destination directly. Since this fabric is configured for Ingress Replication, leaf1 uses its flood list for VLAN 200 to package the broadcast ARP into a unicast VXLAN packet and send it to every VTEP (PE) in the flood list. This includes leaf3.

When leaf3 (the Junos OS leaf) receives this, it decapsulates the VXLAN packet and floods the ARP to all local ports in VLAN 200. h4 now receives this ARP packet, builds its local ARP cache, and sends an ARP reply.

Let's look at this ARP reply:

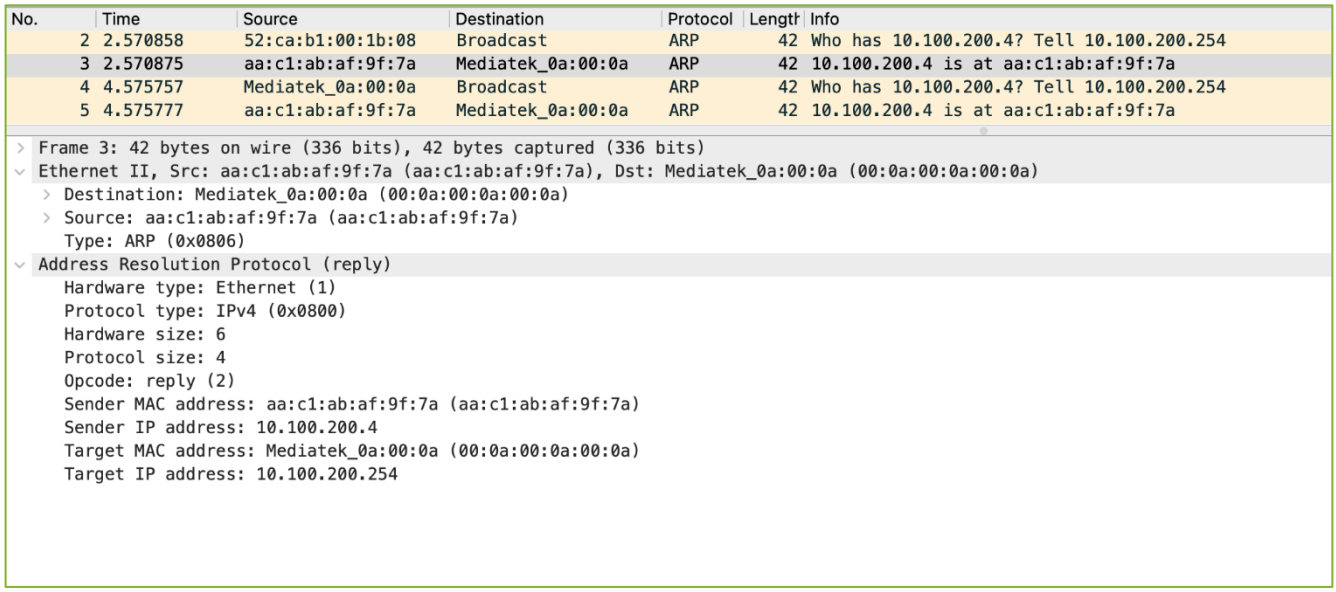


Figure 5: Packet Capture of an ARP Reply

At first glance, this seems like an ordinary ARP reply. If you carefully observe the destination MAC address in the Ethernet header, however, you'll notice that this is the anycast gateway MAC address. When this ARP reply reaches leaf3, it will consume this packet since it owns the anycast MAC address and leaf1 will never see this ARP reply. Because of this, the ARP entry for 10.100.200.4 will always remain incomplete, and leaf1 has no knowledge of how to forward traffic to this destination.

```
leaf1# show ip arp

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       CP - Added via L2RIB, Control plane Adjacencies
       PS - Added via L2RIB, Peer Sync
       RO - Re-Originated Peer Sync Entry
       D - Static Adjacencies attached to down interface

IP ARP Table for context default
Total number of entries: 4
Address         Age       MAC Address     Interface       Flags
198.51.100.1    00:09:48  5295.1e00.1b08  Ethernet1/1
198.51.100.9    00:09:29  520a.3600.1b08  Ethernet1/2
10.100.100.1    00:02:00  aac1.ab3e.c444  Vlan100
10.100.200.4    00:00:03  INCOMPLETE      Vlan200
```

For Asymmetric IRB to work, the VTEP must build its ARP table from EVPN Type-2 MAC+IP routes. Without this functionality, Asymmetric IRB will fail, as demonstrated above. NXOS does not do this, which is why it does not support Asymmetric IRB.

The following packet walk provides an easy-to-understand visual representation of the failure:

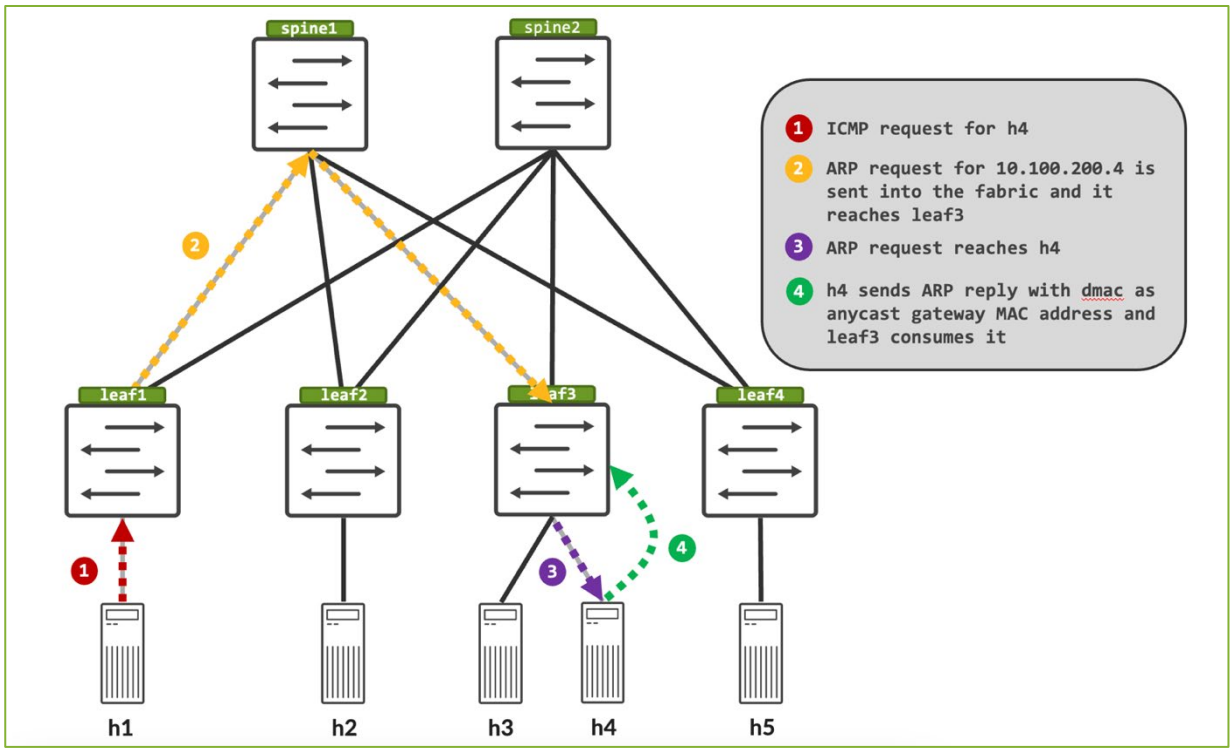


Figure 6: Packet Walk of Asymmetric IRB Failing

Symmetric IRB on NXOS and Asymmetric IRB on Junos OS

Overview and Topology

For this use case, we'll be using the following topology:

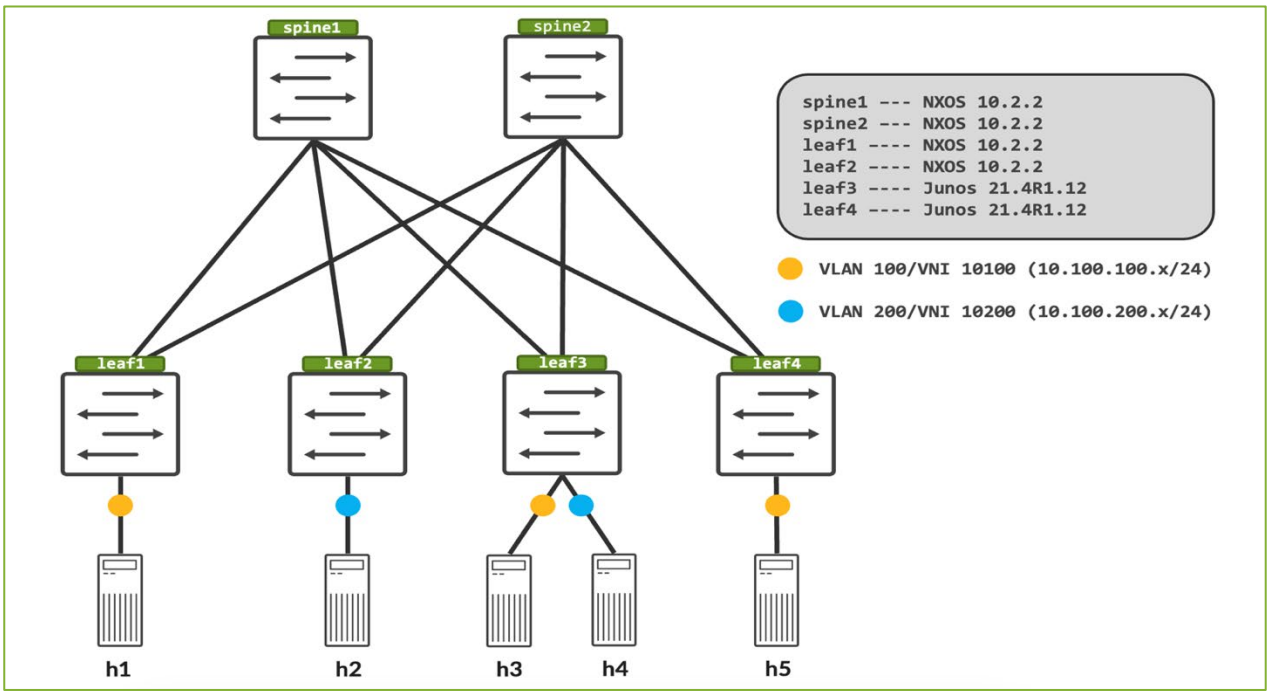


Figure 7: Topology of the Fabric Used in this Document, with Assigned VNIs

Under most circumstances, Symmetric and Asymmetric IRB cannot interoperate with each other. Configured as is (one leaf configured for Symmetric IRB and another leaf configured for Asymmetric IRB), this use case will fail.

NXOS, however, provides a new mode called EVPN Hybrid Mode that allows for Symmetric IRB on NXOS to communicate with any other VTEP (PE) that is configured for Asymmetric IRB. This Hybrid Mode requires additional configuration which essentially converts the NXOS device into Asymmetric IRB mode, with a few finer details that we'll look at here.

Symmetric IRB Between NXOS Leafs

Before diving into how EVPN Hybrid Mode works, let's first consider how Symmetric IRB works between leaf1 and leaf2 (both NXOS). The goal is to get h1 to talk to h2 (communication between VLAN 100 and VLAN 200). On the NXOS leaf switches Symmetric IRB requires the following configuration (sample configuration from leaf1 as an example):

Like before, an anycast gateway MAC address is configured. In addition to the L2VNI, a new L3VNI is created as well and mapped to its respective VLAN.

```
nv overlay evpn
feature bgp
feature fabric forwarding
feature interface-vlan
feature vn-segment-vlan-based
feature lldp
feature nv overlay
!
fabric forwarding anycast-gateway-mac 000a.000a.000a
!
```

```
vlan 1,100,300
vlan 100
  vn-segment 10100
vlan 300
  vn-segment 10300
!
```

What makes the VNI an L3VNI is its association to a VRF (shown below). The VRF is enabled for the IPv4 unicast address family and configured with an import/export route-target for EVPN.

```
vrf context Tenant1
  vni 10300
  rd auto
  address-family ipv4 unicast
    route-target import 65421:300
    route-target import 65421:300 evpn
    route-target export 65421:300
    route-target export 65421:300 evpn
!
```

IRB interfaces are configured and associated to the respective tenant VRF. An IRB interface for the L3VNI is created as well and placed in the same VRF. There is no IP address configured for this and thus '*ip forward*' must be enabled to allow IPv4 traffic over this interface.

```
interface Vlan100
  no shutdown
  vrf member Tenant1
  ip address 10.100.100.254/24
  fabric forwarding mode anycast-gateway
!
interface Vlan300
  no shutdown
  vrf member Tenant1
  ip forward
!
```

The NVE interface is configured for the L2VNI as well as the L3VNI (with the '*associate-vrf*' keyword). The VRF is also defined under BGP, and direct routes are redistributed. This is needed for the IRB subnet route to be advertised via BGP – this enables silent hosts to be discovered in the fabric.

```
route-map allow-loopback permit 10
  match interface loopback0
route-map permit-all permit 10
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
  member vni 10300 associate-vrf
!
router bgp 65421
  router-id 192.0.2.1
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
    maximum-paths 4
  address-family l2vpn evpn
    advertise-pip
  template peer evpn
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 192.0.2.11
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.22
```

```
    inherit peer evpn
    remote-as 65500
  neighbor 198.51.100.1
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.9
    remote-as 65500
    address-family ipv4 unicast
  vrf Tenant1
    address-family ipv4 unicast
      redistribute direct route-map permit-all
evpn
  vni 10100 l2
    rd 192.0.2.1:100
    route-target import 100:100
    route-target export 100:100
```

leaf1 learns h1's MAC address and IPv4 address using ARP/ND. We can initiate a ping from h1 (to its default gateway, 10.100.100.254, which is present on leaf1) to trigger this process.

```
root@h1:~# ping 10.100.100.254
PING 10.100.100.254 (10.100.100.254) 56(84) bytes of data.
64 bytes from 10.100.100.254: icmp_seq=1 ttl=255 time=4.43 ms
64 bytes from 10.100.100.254: icmp_seq=2 ttl=255 time=1.95 ms
64 bytes from 10.100.100.254: icmp_seq=3 ttl=255 time=2.16 ms
64 bytes from 10.100.100.254: icmp_seq=4 ttl=255 time=1.86 ms
^C
--- 10.100.100.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 1.856/2.599/4.431/1.064 ms
```

Within the VRF, an ARP entry is created for h1, and this is pushed to BGP EVPN as well. This gets advertised as an EVPN Type-2 MAC+IP route to the spine switches, and from there to leaf2.

```
leaf1# show ip arp vrf Tenant1

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       CP - Added via L2RIB, Control plane Adjacencies
       PS - Added via L2RIB, Peer Sync
       RO - Re-Originated Peer Sync Entry
       D - Static Adjacencies attached to down interface

IP ARP Table for context Tenant1
Total number of entries: 1
Address         Age       MAC Address     Interface       Flags
10.100.100.1    00:01:33  aac1.ab3e.c444  Vlan100

leaf1# show bgp l2vpn evpn 10.100.100.1
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.ab3e.c444]:[32]:[10.100.100.1]/272, version 19
Paths: (1 available, best #1)
Flags: (0x000102) (high32 00000000) on xmit-list, is not in l2rib/evpn

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    192.0.2.1 (metric 0) from 0.0.0.0 (192.0.2.1)
      Origin IGP, MED not set, localpref 100, weight 32768
      Received label 10100 10300
      Extcommunity: RT:100:100 RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Path-id 1 advertised to peers:
    192.0.2.11         192.0.2.22
```

The extended communities added to this are important. We can see both the L2 route-targets, the L3VNI route-target, along with the router MAC address. This router MAC address corresponds to IRB interface MAC address, while the L3VNI route-target is used to identify (and import into) the customer VRF:

```
leaf1# show interface vlan100
Vlan100 is up, line protocol is up, autostate enabled
  Hardware is EtherSVI, address is  52ca.b100.1b08
  Internet Address is 10.100.100.254/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
   reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  ARP type: ARPA
  Last clearing of "show interface" counters never
  L3 in Switched:
    ucast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes
```

On leaf2, the route gets imported into the VRF table with a matching L3VNI route-target.

```
leaf2# show bgp l2vpn evpn 10.100.100.1
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.ab3e.c444]:[32]:[10.100.100.1]/272, version 21
Paths: (2 available, best #2)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.22 (192.0.2.22)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100 10300
      Extcommunity: RT:100:100 RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: Tenant1 L3-10300
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100 10300
      Extcommunity: RT:100:100 RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Path-id 1 not advertised to any peer

Route Distinguisher: 192.0.2.2:3    (L3VNI 10300)
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.ab3e.c444]:[32]:[10.100.100.1]/272, version 20
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported from 192.0.2.1:100:[2]:[0]:[0]:[48]:[aac1.ab3e.c444]:[32]:[10.100.100.1]/272
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100 10300
      Extcommunity: RT:100:100 RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Path-id 1 not advertised to any peer
```

This adds a host route (a /32 route) entry into VRF table:

```
leaf2# show ip route 10.100.100.1 vrf Tenant1
IP Route Table for VRF "Tenant1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.100.100.1/32, ubest/mbest: 1/0
    *via 192.0.2.1%default, [20/0], 3d04h, bgp-65422, external, tag 65500, segid: 10300 tunnelid:
0xc0000201 encap: VXLAN
```

In addition to the host route, an EVPN Type-5 route (for the IRB subnet) is also advertised in accordance with the RFC (RFC 9135) for silent hosts (to trigger the gleaning process for such hosts). This is done via the '*redistribute direct*' command under the VRF defined in BGP. This subnet route is pulled into the VRF table as well:

```
leaf2# show bgp l2vpn evpn 10.100.100.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:3
BGP routing table entry for [5]:[0]:[0]:[24]:[10.100.100.0]/224, version 14
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
  Gateway IP: 0.0.0.0
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.22 (192.0.2.22)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 10300
      Extcommunity: RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: Tenant1 L3-10300
  Gateway IP: 0.0.0.0
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 10300
      Extcommunity: RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Path-id 1 not advertised to any peer

Route Distinguisher: 192.0.2.2:3    (L3VNI 10300)
BGP routing table entry for [5]:[0]:[0]:[24]:[10.100.100.0]/224, version 10
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported from 192.0.2.1:3:[5]:[0]:[0]:[24]:[10.100.100.0]/224
  Gateway IP: 0.0.0.0
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin incomplete, MED not set, localpref 100, weight 0
      Received label 10300
      Extcommunity: RT:65421:300 ENCAP:8 Router MAC:52ca.b100.1b08

  Path-id 1 not advertised to any peer

leaf2# show ip route 10.100.100.0/24 vrf Tenant1
IP Route Table for VRF "Tenant1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.100.100.0/24, ubest/mbest: 1/0
    *via 192.0.2.1%default, [20/0], 3d04h, bgp-65422, external, tag 65500, segid: 10300 tunnelid:
0xc0000201 encap: VXLAN
```

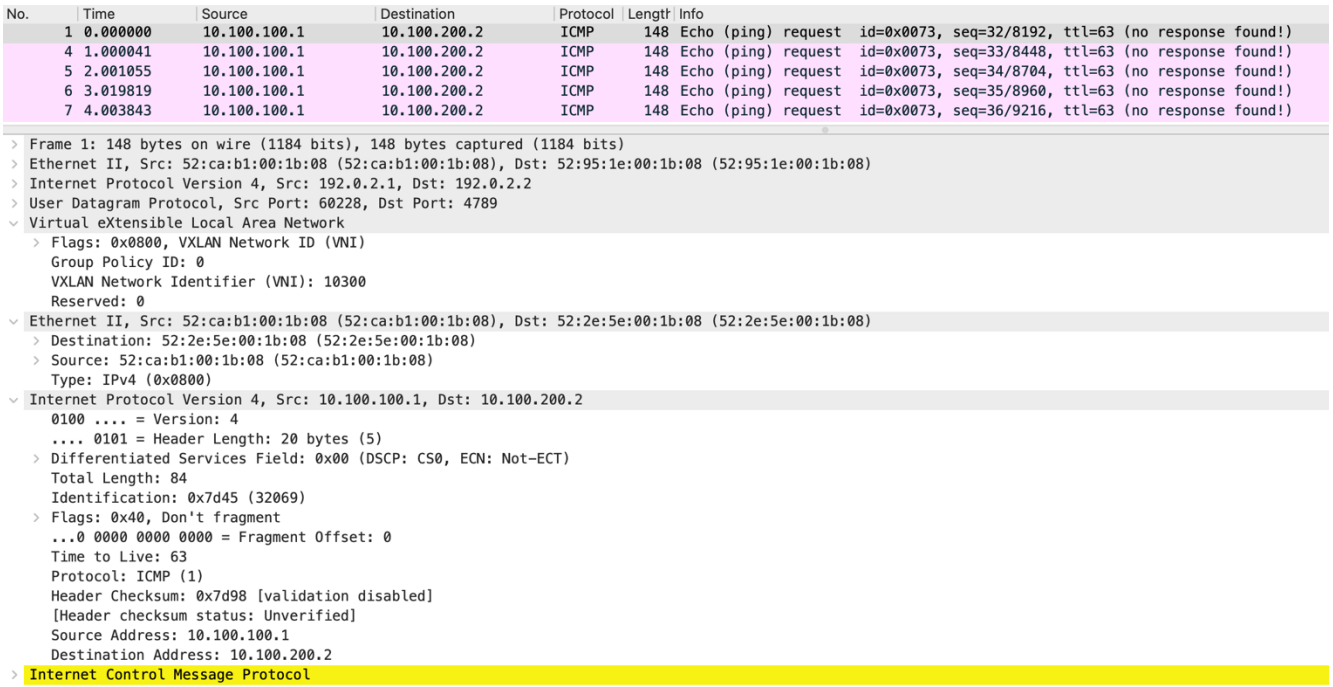
A similar process happens in reverse for h2. On leaf1, h2s host route and a corresponding subnet route should exist in the VRF table:

```
leaf1# show ip route vrf Tenant1
IP Route Table for VRF "Tenant1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.100.100.0/24, ubest/mbest: 1/0, attached
```

```
    *via 10.100.100.254, Vlan100, [0/0], 3d04h, direct
10.100.100.1/32, ubest/mbest: 1/0, attached
    *via 10.100.100.1, Vlan100, [190/0], 00:31:24, hmm
10.100.100.254/32, ubest/mbest: 1/0, attached
    *via 10.100.100.254, Vlan100, [0/0], 3d04h, local
10.100.200.0/24, ubest/mbest: 1/0
    *via 192.0.2.2%default, [20/0], 3d04h, bgp-65421, external, tag 65500, segid: 10300 tunnelid:
0xc0000202 encap: VXLAN

10.100.200.2/32, ubest/mbest: 1/0
    *via 192.0.2.2%default, [20/0], 3d04h, bgp-65421, external, tag 65500, segid: 10300 tunnelid:
0xc0000202 encap: VXLAN
```

h1 should now be able to ping h2 and a data-plane capture shows that the VNI in the VXLAN header is the L3VNI:

```
root@h1:~# ping 10.100.200.2
PING 10.100.200.2 (10.100.200.2) 56(84) bytes of data.
64 bytes from 10.100.200.2: icmp_seq=1 ttl=62 time=19.5 ms
64 bytes from 10.100.200.2: icmp_seq=2 ttl=62 time=29.8 ms
64 bytes from 10.100.200.2: icmp_seq=3 ttl=62 time=12.0 ms
64 bytes from 10.100.200.2: icmp_seq=4 ttl=62 time=12.0 ms
64 bytes from 10.100.200.2: icmp_seq=5 ttl=62 time=22.6 ms
^C
--- 10.100.200.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 11.968/19.167/29.789/6.750 ms
```



Configuring Asymmetric IRB on Junos OS

The next step is to configure leaf3 for Asymmetric IRB. For this, we require that the IRB interface for both VLAN100 and VLAN200 exist on leaf3, along with their corresponding L2VNIs. Routing-instances are created for both VLAN100 and VLAN200 with service-type VLAN-based. As we've seen this configuration before, explanation is not provided again here.

```
admin@leaf3# show
interfaces {

*snip*

    irb {
```

```
        unit 100 {
            virtual-gateway-accept-data;
            family inet {
                address 10.100.100.252/24 {
                    virtual-gateway-address 10.100.100.254;
                }
            }
            virtual-gateway-v4-mac 00:0a:00:0a:00:0a;
        }
        unit 200 {
            virtual-gateway-accept-data;
            family inet {
                address 10.100.200.252/24 {
                    virtual-gateway-address 10.100.200.254;
                }
            }
            virtual-gateway-v4-mac 00:0a:00:0a:00:0a;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.0.2.3/32;
            }
        }
    }
}
policy-options {
    policy-statement ECMP {
        then {
            load-balance per-flow;
        }
    }
    policy-statement allow-loopback {
        from interface lo0.0;
        then accept;
    }
}
routing-instances {
    v100_mac_vrf {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway do-not-advertise;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        interface xe-0/0/2.0;
        route-distinguisher 192.0.2.3:100;
        vrf-target target:100:100;
        vlans {
            v100 {
                vlan-id 100;
                l3-interface irb.100;
                vxlan {
                    vni 10100;
                }
            }
        }
    }
    v200_mac_vrf {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway do-not-advertise;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        route-distinguisher 192.0.2.3:200;
```

```
        vrf-target target:200:200;
        vlans {
            v200 {
                vlan-id 200;
                l3-interface irb.200;
                vxlan {
                    vni 10200;
                }
            }
        }
    }
}
routing-options {
    router-id 192.0.2.3;
    autonomous-system 65423;
    forwarding-table {
        export ECMP;
    }
}
protocols {
    bgp {
        group underlay {
            type external;
            family inet {
                unicast;
            }
            export allow-loopback;
            peer-as 65500;
            multipath;
            neighbor 198.51.100.5;
            neighbor 198.51.100.13;
        }
        group overlay {
            type external;
            multihop;
            local-address 192.0.2.3;
            family evpn {
                signaling;
            }
            peer-as 65500;
            neighbor 192.0.2.11;
            neighbor 192.0.2.22;
        }
    }
}
```

The IRB interfaces come up if there is at least one remote VXLAN tunnel that is created (these tunnels are created exchanging EVPN Type-3 IMET routes):

```
admin@leaf3> show ethernet-switching vxlan-tunnel-end-point remote
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   192.0.2.3        lo0.0    0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 192.0.2.1        v100_mac_vrf             556       vtep.32770   1788    RNVE
    VNID          MC-Group-IP
    10100         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 192.0.2.2        v200_mac_vrf             576       vtep.32771   1789    RNVE
    VNID          MC-Group-IP
    10200         0.0.0.0

admin@leaf3> show interfaces irb
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 640, SNMP ifIndex: 506
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 02:05:86:71:49:00, Hardware address: 02:05:86:71:49:00
  Last flapped   : Never
    Input packets : 0
```

```
    Output packets: 0

  Logical interface irb.100 (Index 567) (SNMP ifIndex 540)
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    Virtual Gateway V4 MAC: 00:0a:00:0a:00:0a
    Bandwidth: 1Gbps
    Routing Instance: v100_mac_vrf Bridging Domain: v100
    Input packets : 0
    Output packets: 8
    Protocol inet, MTU: 1500
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop cnt: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.100.100/24, Local: 10.100.100.252, Broadcast: 10.100.100.255
        Destination: 10.100.100/24, Local: 10.100.100.254, Broadcast: 10.100.100.255

  Logical interface irb.200 (Index 568) (SNMP ifIndex 541)
    Flags: Up SNMP-Traps 0x4000 Encapsulation: ENET2
    Virtual Gateway V4 MAC: 00:0a:00:0a:00:0a
    Bandwidth: 1Gbps
    Routing Instance: v200_mac_vrf Bridging Domain: v200
    Input packets : 0
    Output packets: 2
    Protocol inet, MTU: 1514
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop cnt: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.100.200/24, Local: 10.100.200.252, Broadcast: 10.100.200.255
        Destination: 10.100.200/24, Local: 10.100.200.254, Broadcast: 10.100.200.255
```

Validating Host to Host Connectivity and Why it Fails

Our primary test is to check connectivity between h2 and h3 since they are in different subnets - h2 is in VLAN200 and h3 is in VLAN100.

h3 can ping its default gateway, 10.100.100.254, which exists on leaf3 as since it is an anycast gateway.

```
root@h3:~# ping 10.100.100.254
PING 10.100.100.254 (10.100.100.254) 56(84) bytes of data.
64 bytes from 10.100.100.254: icmp_seq=1 ttl=64 time=119 ms
64 bytes from 10.100.100.254: icmp_seq=2 ttl=64 time=114 ms
64 bytes from 10.100.100.254: icmp_seq=3 ttl=64 time=128 ms
64 bytes from 10.100.100.254: icmp_seq=4 ttl=64 time=149 ms
^C
--- 10.100.100.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 113.884/127.348/149.132/13.538 ms
```

However, h3 cannot ping h2:

```
root@h3:~# ping 10.100.200.2
PING 10.100.200.2 (10.100.200.2) 56(84) bytes of data.
^C
--- 10.100.200.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 90ms
```

The above fails because, while leaf3 has the required information for h2 (it has an IRB interface for VLAN200 and is thus able to ARP for h2 directly), leaf2 does not have any information about h3 in its forwarding table. It is receiving the EVPN Type-2 MAC+IP route but it does not know what to do with it, since there is no corresponding L2VNI (more importantly, there is no matching import route-target). As a result, leaf3 simply drops the prefix, as seen below:

```
*snip*

leaf2# show bgp event-history prefixes
```

```
2022-12-17T13:49:54.428360000+00:00 [M 27] [bgp] E_DEBUG [bgp_af_process_nlri:7447] (default) PFX:
[L2VPN EVPN] Dropping prefix [2]:[0]:[0]:[48]:[aac1.abc1.c674]:[32]:[10.100.100.3]/144 from peer
192.0.2.22, due to attribu
te policy rejected
2022-12-17T13:49:54.428358000+00:00 [M 27] [bgp] E_DEBUG [bgp_af_process_nlri:7447] (default) PFX:
[L2VPN EVPN] Dropping prefix [2]:[0]:[0]:[48]:[aac1.abc1.c674]:[0]:[0.0.0.0]/112 from peer 192.0.2.22,
due to attribute pol
icy rejected

*snip*
```

Introducing EVPN Hybrid Mode on NXOS

The mismatch between the two approaches is resolved by NXOS' EVPN Hybrid Mode. Cisco's documentation on Hybrid Mode functionality helps to convert the leaf into Asymmetric IRB operation, despite this not being supported outside of hybrid mode by NXOS. Let's examine this in more detail:

From Cisco's documentation, the most important information about Hybrid Mode is:

- NX-OS symmetric IRB VTEPs must be provisioned with all subnets in an IP VRF that are stretched to asymmetric VTEPs in the fabric.
- NX-OS symmetric IRB VTEPs must be provisioned with subnets in an IP VRF that are stretched to asymmetric VTEPs in "hybrid" mode using "fabric forwarding mode anycast-gateway hybrid" CLI under the subnet SVI interface.
- All symmetric IRB VTEPs must have the hybrid mode enabled when interoperating with asymmetric VTEPs in each fabric.

In other words, NXOS' Hybrid Mode enables Asymmetric IRB support, but with the continued existence of the IP VRF and the corresponding L3VNI. This is the additional configuration that we're going to add on leaf2 now:

- Create VLAN100 (the L2 VLAN) and corresponding L2VNI (VNI 10100).
- Create an IRB interface for VLAN100 and enable it for EVPN Hybrid Mode.
- Add this new VNI to the NVE interface (nve1).
- Add this new L2VNI to EVPN.

Configuring EVPN Hybrid Mode on NXOS

For EVPN Hybrid Mode, we'll create the required VLANs for Asymmetric IRB (VLAN 100, in this case) and all the necessary configuration for Asymmetric IRB to work for this VLAN/VNI. In addition, the fabric forwarding mode is configured as '*anycast-gateway hybrid'* under the IRB interface.

```
vlan 100
  vn-segment 10100
!
interface Vlan100
  no shutdown
  vrf member Tenant1
  ip address 10.100.100.254/24
  fabric forwarding mode anycast-gateway hybrid
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
  member vni 10200
    ingress-replication protocol bgp
  member vni 10300 associate-vrf
!
evpn
  vni 10100 l2
    rd 192.0.2.2:100
    route-target import 100:100
    route-target export 100:100
```

```
  vni 10200 l2
    rd 192.0.2.2:200
    route-target import 200:200
    route-target export 200:200
```

Understanding How EVPN Hybrid Mode Interoperates Fixes the Problem

As the L2VNI now exists on leaf2, the EVPN Type-2 MAC+IP route is accepted.

```
leaf2# show bgp l2vpn evpn 10.100.100.3
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.2:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.abc1.c674]:[32]:[10.100.100.3]/248, version 222
Paths: (1 available, best #1)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.3:100:[2]:[0]:[0]:[48]:[aac1.abc1.c674]:[32]:[10.100.100.3]/248
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.22 (192.0.2.22)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8

  Path-id 1 not advertised to any peer

Route Distinguisher: 192.0.2.3:100
BGP routing table entry for [2]:[0]:[0]:[48]:[aac1.abc1.c674]:[32]:[10.100.100.3]/248, version 215
Paths: (2 available, best #2)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: L2-10100
  AS-Path: 65500 65423 , path sourced external to AS
    192.0.2.3 (metric 0) from 192.0.2.22 (192.0.2.22)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8

  Path-id 1 not advertised to any peer
```

This is installed into the MAC address table via the same process that we looked at earlier in the bridged overlay section.

```
leaf2# show mac address-table address aac1.abc1.c674
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
        (NA)- Not Applicable
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C  100     aac1.abc1.c674   dynamic  NA         F      F    nve1(192.0.2.3)
```

The above configuration, on leaf2, should now allow it to function as Asymmetric IRB (only for destinations behind leaf switches doing Asymmetric IRB). As discussed previously in this document, however, Asymmetric IRB did not work on NXOS due to differences in ARP process functionality. Let's test to see if this has been resolved.

```
leaf2# show ip arp vrf Tenant1

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       CP - Added via L2RIB, Control plane Adjacencies
       PS - Added via L2RIB, Peer Sync
       RO - Re-Originated Peer Sync Entry
       D - Static Adjacencies attached to down interface

IP ARP Table for context Tenant1
Total number of entries: 1
Address         Age       MAC Address     Interface       Flags
10.100.200.2    00:00:08  aac1.ab83.9993  Vlan200
```

Note that there is no ARP for 10.100.100.3. Despite this, h3 can ping h2 when the switches are configured in this fashion, so how is it working?

```
root@h3:~# ping 10.100.200.2
PING 10.100.200.2 (10.100.200.2) 56(84) bytes of data.
64 bytes from 10.100.200.2: icmp_seq=1 ttl=63 time=129 ms
64 bytes from 10.100.200.2: icmp_seq=2 ttl=63 time=276 ms
64 bytes from 10.100.200.2: icmp_seq=3 ttl=63 time=148 ms
64 bytes from 10.100.200.2: icmp_seq=4 ttl=63 time=127 ms
^C
--- 10.100.200.2 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 11ms
rtt min/avg/max/mdev = 127.399/169.990/275.513/61.443 ms
```

The answer is the '*fabric forwarding mode anycast-gateway hybrid'* command. Looking at the l2route table, you can see that the 'Asymmetric' flag is set (Asy), and the IP-MAC binding is sent directly to adjacency manager (AM) instead of installing it in the ARP table:

```
leaf2# show l2route evpn mac-ip all detail
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv(D):Del Pending (S):Stale (C):Clear
(Ps):Peer Sync (Ro):Re-Originated (Orp):Orphan (Asy):Asymmetric (Gw):Gateway
(Piporp): Directly connected Orphan to PIP based vPC BGW
(Pipporp): Orphan connected to peer of PIP based vPC BGW
Topology    Mac Address    Host IP                                   Prod   Flags         Seq
No     Next-Hops
----------- -------------- ----------------------------------------- ------ ---------- ---------- -------
--------------------------------
100         aac1.ab3e.c444 10.100.100.1                              BGP    -
-            0         192.0.2.1 (Label: 10100)
            Sent To: AM
            encap-type:1
100         aac1.abc1.c674
10.100.100.3                            BGP    Asy           0         192.0.2.3 (Label:
10100)
            Sent To: AM
            Peer ID: 2
            encap-type:1
100         000a.000a.000a
10.100.100.254                          BGP    Stt,Asy       0         192.0.2.3 (Label:
10100)
            Sent To: AM
            ESI : 0500.00ff.8f00.0027.7400
            encap-type:1
200         aac1.ab83.9993
10.100.200.2                            HMM    L,            0         Local

            L3-Info: 10300
            Sent To: BGP
200         000a.000a.000a
10.100.200.254                          BGP    Stt,Asy       0         192.0.2.3 (Label:
10200)
            ESI : 0500.00ff.8f00.0027.d800
            encap-type:1
```

We can confirm this from the logs below. BGP EVPN sends this to l2rib, which in turn sends it to AM.

```
leaf2# show system internal l2rib event-history mac-ip
2022-12-17T15:08:59.587921000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_svr_mac_ip_ent_gpb_encode:587]
(100,aac1.abc1.c674,10.100.100.3,5): Encoding MAC-IP best route (ADD, client id 0), esi: (F)
2022-12-17T15:08:59.583040000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_obj_mac_ip_route_create:1434]
(100,aac1.abc1.c674,10.100.100.3,5):  ESI: (F), port-channel ifIndex: 0
2022-12-17T15:08:59.582824000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_obj_mac_ip_route_create:1422]
(100,aac1.abc1.c674,10.100.100.3,5):  adminDist: 20, SOO: 0, peerID: 2, peer ifIndex: 1191182338
2022-12-17T15:08:59.582823000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_obj_mac_ip_route_create:1410]
(100,aac1.abc1.c674,10.100.100.3,5): MAC-IP route created with flags: 32, L3 VNI: 0, seqNum: 0
2022-12-17T15:08:59.582812000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_obj_mac_ip_create:208]
(100,aac1.abc1.c674,10.100.100.3): MAC-IP entry created
2022-12-17T15:08:59.582452000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_client_show_mac_ip_route_msg:1462]
NH: 192.0.2.3 (Label: 10100)
2022-12-17T15:08:59.582450000+00:00 [M 27] [l2rib] E_DEBUG [8381]: Rcvd MAC-IP ROUTE msg: res 0, esi
(F), es_type 0, tag 0, ifindex 0, nh_count 1, pc-ifindex 0
2022-12-17T15:08:59.582447000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_client_show_mac_ip_route_msg:1444]
Rcvd MAC-IP ROUTE msg: flags Asy, admin_dist 0, seq 0, soo 0, peerid 0,
2022-12-17T15:08:59.582445000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_client_show_mac_ip_route_msg:1436]
Rcvd MAC-IP ROUTE msg: (100, aac1.abc1.c674, 10.100.100.3), vrf_id 0, encap_type 1,
2022-12-17T15:08:59.582443000+00:00 [M 27] [l2rib] E_DEBUG [l2rib_client_show_mac_ip_route_msg:1429]
Rcvd MAC-IP ROUTE msg: (100, aac1.abc1.c674, 10.100.100.3), l2 vni 0, l3 vni 0

*snip*

leaf2# show system internal adjmgr internal event-history events
2022-12-17T15:08:59.683335000+00:00 [M 27] [adjmgr] E_DEBUG AM upd do work: Event=sentRouteToURIB,
AFI=IPv4, routeCount=1
2022-12-17T15:08:59.683219000+00:00 [M 27] [adjmgr] E_DEBUG Append IPv4 adj route add to UPD:
Result=Success, IP=10.100.100.3, IOD=77, Interface=Vlan100, prot0PhyIod=76, prot0PhyInterface=nve-
peer2, prot1PhyIod=0, prot1Phy
Interface=None, tableId=0x3, adminDistance=250, mobility=0, nhCount=0
2022-12-17T15:08:59.683212000+00:00 [M 27] [adjmgr] E_DEBUG Append adj prot info to UPD:
Result=Success, protAdjIndex=0, phyifIndex=0x47000002, phyIfType=71, Encap=1, tunnelId=0xffffffff
2022-12-17T15:08:59.614494000+00:00 [M 27] [adjmgr] E_DEBUG Processed L2RIB Msg.
dbgStr=##48,5c,1b,1,1b,10,33,35,36,15,7a,c,67,12,68,9,44,b
2022-12-17T15:08:59.595194000+00:00 [M 27] [adjmgr] E_DEBUG Add to UPD work: Result=Success,
IP=10.100.100.3, IOD=77, Interface=Vlan100, AFI=IPv4, workBits=0x1
2022-12-17T15:08:59.589259000+00:00 [M 27] [adjmgr] E_DEBUG Received MAC-IP update with AFI: 2 IP:
10.100.100.3 VRF: 0x0 l2r_ifIdx: 1191182338 peerId: 0x2 seqNum: 0 flags: 0x7 MAC : aac1.abc1.c674
2022-12-17T15:08:59.589230000+00:00 [M 27] [adjmgr] E_DEBUG Processed L2RIB Msg. dbgStr=##48,5d

*snip*
```

Our final confirmation is that the entry exists in the adjacency table:

```
leaf2# show ip adjacency vrf Tenant1

Flags: # - Adjacencies Throttled for Glean
       G - Adjacencies of vPC peer with G/W bit
       R - Adjacencies learnt remotely
       CP - Added via L2RIB, Control plane Adjacencies
       PS - Added via L2RIB, Peer Sync
       RO - Re-Originated Peer Sync Entry
       CC - Consistency check pending

IP Adjacency Table for VRF Tenant1
Total number of entries: 2
Address         MAC Address     Pref Source     Interface         Mobility Flags
10.100.200.2    aac1.ab83.9993  50   arp        Vlan200
10.100.100.3    aac1.abc1.c674  50   am_l2rib   Vlan100                 0 CP R
```

As you can see, the source for 10.100.100.3 is ‘am_l2rib’ and the 'CP' flag is set, which implies it was learnt via L2RIB.

It should be noted that the above is compliant with the relevant RFC. There is no requirement in RFC9135 to see the entry in the ARP table. RFC9135 states the following:

"*For IP-to-MAC bindings learned via EVPN, an implementation may choose to import these bindings directly to the respective forwarding table (such as an adjacency/next-hop table) as opposed to importing them to ARP or ND protocol tables*"

Currently on NXOS devices Asymmetric IRB can ONLY work with ‘fabric forwarding mode anycast-gateway hybrid’ under the IRB interfaces. Outside of this it continues to be unsupported on NXOS. This command allows NXOS to tie Asymmetric into Symmetric IRB and L3VNIs and create checks to ensure that Asymmetric IRB functions only when Symmetric IRB with L3VNIs is enabled.

Symmetric IRB on NXOS and Symmetric IRB on Junos OS

Overview and Topology

Symmetric IRB requires only the necessary VLANs and VNIs to exist on the leaf switches and uses a common L3VNI to stitch the subnets together. The L3VNI has a 1:1 mapping to an IP VRF that is created for the customer.

The same topology is used as the previous section, with the only difference being the Junos OS version, which is 22.2R2, in this case.

Configuring Symmetric IRB on NXOS

As we've seen how Symmetric IRB is configured on NXOS in the previous section, the explanation is not provided again here. A sample configuration for a leaf is below:

```
nv overlay evpn
feature bgp
feature fabric forwarding
feature interface-vlan
feature vn-segment-vlan-based
feature lldp
feature nv overlay
!
fabric forwarding anycast-gateway-mac 000a.000a.000a
!
vlan 1,100,300
vlan 100
  vn-segment 10100
vlan 300
  vn-segment 10300
!
route-map allow-loopback permit 10
  match interface loopback0
route-map permit-all permit 10
!
vrf context Tenant1
  vni 10300
  rd 192.0.2.1:1
  address-family ipv4 unicast
    route-target import 300:300
    route-target import 300:300 evpn
    route-target export 300:300
    route-target export 300:300 evpn
!
interface Vlan100
  no shutdown
  vrf member Tenant1
  ip address 10.100.100.254/24
  fabric forwarding mode anycast-gateway
!
interface Vlan300
  no shutdown
  vrf member Tenant1
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  advertise virtual-rmac
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
  member vni 10300 associate-vrf
!
interface Ethernet1/3
  switchport access vlan 100
!
router bgp 65421
```

```
  router-id 192.0.2.1
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
    maximum-paths 4
  address-family l2vpn evpn
    advertise-pip
  template peer evpn
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 192.0.2.11
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.22
    inherit peer evpn
    remote-as 65500
  neighbor 198.51.100.1
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.9
    remote-as 65500
    address-family ipv4 unicast
  vrf Tenant1
    address-family ipv4 unicast
      redistribute direct route-map permit-all
evpn
  vni 10100 l2
    rd 192.0.2.1:100
    route-target import 100:100
    route-target export 100:100
```

Configuring Symmetric IRB on Junos OS

For Junos OS, let's focus on a few specifics and break down what we are doing with the configuration. IRB interfaces are configured like before, but since this is Symmetric IRB only the local IRB interfaces are needed. This means that for leaf3, we need only VLAN 200's IRB interface.

```
Interfaces {
    irb {
        unit 200 {
            virtual-gateway-accept-data;
            family inet {
                address 10.100.200.252/24 {
                    virtual-gateway-address 10.100.200.254;
                }
            }
            virtual-gateway-v4-mac 00:0a:00:0a:00:0a;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.0.2.3/32;
            }
        }
    }
}
```

A routing-instance of instance-type mac-vrf is created for the L2 VLAN. This is configured with the necessary EVPN parameters.

Another routing-instance is created for the L3 VRF (which will be the tenant VRF) and Symmetric IRB is enabled under this using '*irb-symmetric-routing'*. The L3VNI must be defined under this.

Remember, by default the subnet route is not advertised – this is done via '*ip-prefix-routing'*. This requires some mandatory parameters – the encapsulation must be set and a VNI must be defined (which needs to match the L3VNI configured earlier).

Finally, the IRB interface needs to be associated to the VRF and a route-distinguisher and route-target must be defined as well.

```
Forwarding-options {
    evpn-vxlan {
        shared-tunnels;
    }
}
routing-instances {
    Tenant1 {
        instance-type vrf;
        protocols {
            evpn {
                irb-symmetric-routing {
                    vni 10300;
                }
                ip-prefix-routes {
                    advertise direct-nexthop;
                    encapsulation vxlan;
                    vni 10300;
                }
            }
        }
        interface irb.200;
        route-distinguisher 192.0.2.3:1;
        vrf-target target:300:300;
    }
    v200_mac_vrf {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway do-not-advertise;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        interface xe-0/0/1.0;
        route-distinguisher 192.0.2.3:200;
        vrf-import v100_accept;
        vrf-target target:200:200;
        vlans {
            v200 {
                vlan-id 200;
                l3-interface irb.200;
                vxlan {
                    vni 10200;
                }
            }
        }
    }
}
routing-options {
    router-id 192.0.2.3;
    autonomous-system 65423;
    forwarding-table {
        export ECMP;
    }
}
protocols {
    bgp {
        group underlay {
            type external;
            family inet {
                unicast;
            }
            export allow-loopback;
            peer-as 65500;
```

```
            multipath;
            neighbor 198.51.100.5;
        }
        group overlay {
            type external;
            multihop;
            local-address 192.0.2.3;
            family evpn {
                signaling;
            }
            peer-as 65500;
            neighbor 192.0.2.11;
        }
    }
}
```

Validating Control Plane and Host to Host Connectivity

Once the MAC address of h3 is learned on leaf3, it is installed in the Ethernet-table and the EVPN table for that MAC VRF.

```
root@leaf3> show ethernet-switching table

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 1 entries, 1 learned
Routing instance : v200_mac_vrf
   Vlan                MAC                 MAC      Logical                 SVLBNH/      Active
   name                address             flags    interface               VENH Index   source
   v200                64:c3:d6:60:75:bd   D        xe-0/0/1.0

root@leaf3> show route table v200_mac_vrf.evpn.0

v200_mac_vrf.evpn.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:192.0.2.3:200::0::00:0a:00:0a:00:0a/304 MAC/IP
                   *[EVPN/170] 23:39:37
                       Indirect
2:192.0.2.3:200::0::64:c3:d6:60:75:bd/304 MAC/IP
                   *[EVPN/170] 22:47:30
                       Indirect
2:192.0.2.3:200::0::00:0a:00:0a:00:0a::10.100.200.254/304 MAC/IP
                   *[EVPN/170] 23:39:37
                       Indirect
2:192.0.2.3:200::0::64:c3:d6:60:75:bd::10.100.200.3/304 MAC/IP
                   *[EVPN/170] 22:47:30
                       Indirect
3:192.0.2.3:200::0::192.0.2.3/248 IM
                   *[EVPN/170] 23:39:37
                       Indirect
```

A detailed look at the EVPN route for the MAC address (and the MAC+IP route) shows that while the MAC only route has the L2VNI and the MAC VRF route-target attached to it, the MAC+IP route has the L3VNI and the IP VRF route-target attached to it. This is important because the IP VRF route-target is how the customer VRF is identified and eventually used to import into the VRF table of a remote leaf.

```
root@leaf3> show route table v200_mac_vrf.evpn.0 evpn-mac-address 64:c3:d6:60:75:bd extensive

v200_mac_vrf.evpn.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
2:192.0.2.3:200::0::64:c3:d6:60:75:bd/304 MAC/IP (1 entry, 1 announced)
        *EVPN   Preference: 170
                Next hop type: Indirect, Next hop index: 0
                Address: 0x74c2714
                Next-hop reference count: 15, key opaque handle: 0x0, non-key opaque handle: 0x0
                Protocol next hop: 192.0.2.3
                Indirect next hop: 0x0 - INH Session ID: 0
                State: <Active Int Ext>
                Age: 22:50:49
                Validation State: unverified
```

```
                Task: v200_mac_vrf-evpn
                Announcement bits (1): 2-rt-export
                AS path: I
                Communities: encapsulation:vxlan(0x8)
                Route Label: 10200
                ESI: 00:00:00:00:00:00:00:00:00:00
                Thread: junos-main

2:192.0.2.3:200::0::64:c3:d6:60:75:bd::10.100.200.3/304 MAC/IP (1 entry, 1 announced)
        *EVPN   Preference: 170
                Next hop type: Indirect, Next hop index: 0
                Address: 0x74c2714
                Next-hop reference count: 15, key opaque handle: 0x0, non-key opaque handle: 0x0
                Protocol next hop: 192.0.2.3
                Indirect next hop: 0x0 - INH Session ID: 0
                State: <Active Int Ext>
                Age: 22:50:49
                Validation State: unverified
                Task: v200_mac_vrf-evpn
                Announcement bits (1): 2-rt-export
                AS path: I
                Communities: target:300:300 encapsulation:vxlan(0x8) router-mac:94:f7:ad:94:d3:40
                Route Label: 10200
                Route Label: 10300
                ESI: 00:00:00:00:00:00:00:00:00:00
                Thread: junos-main
```

Junos OS is also advertising an EVPN Type-5 subnet route for the 10.100.200.0/24 subnet since IRB.200 is attached to this IP VRF. This is within the EVPN table for the IP VRF and includes the IP VRF route-target, same as the EVPN Type-2 MAC+IP route as explained earlier.

This EVPN Type-5 route is a pure Type-5 route; it does not have any overlay gateway address. Thus requiring no recursive lookup resolution for the overlay index.

```
root@leaf3> show route table Tenant1.evpn.0 extensive

*snip*

5:192.0.2.3:1::0::10.100.200.0::24/248 (1 entry, 1 announced)
        *EVPN   Preference: 170
                Next hop type: Fictitious, Next hop index: 0
                Address: 0x74c25c4
                Next-hop reference count: 2, key opaque handle: 0x0, non-key opaque handle: 0x0
                Next hop:
                State: <Active Int Ext>
                Age: 23:51:47
                Validation State: unverified
                Task: Tenant1-EVPN-L3-context
                Announcement bits (1): 1-rt-export
                AS path: I
                Communities: encapsulation:vxlan(0x8) router-mac:94:f7:ad:94:d3:40
                Route Label: 10300
                Overlay gateway address: 0.0.0.0
                ESI 00:00:00:00:00:00:00:00:00:00
                Thread: junos-main
```

On the NXOS leaf, these are imported into the VRF table since there is a matching route-target import.

```
Tme-nexus01# show ip route vrf Tenant1
IP Route Table for VRF "Tenant1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.100.100.0/24, ubest/mbest: 1/0, attached
    *via 10.100.100.254, Vlan100, [0/0], 1d00h, direct
10.100.100.1/32, ubest/mbest: 1/0, attached
    *via 10.100.100.1, Vlan100, [190/0], 1d00h, hmm
10.100.100.254/32, ubest/mbest: 1/0, attached
    *via 10.100.100.254, Vlan100, [0/0], 1d00h, local
10.100.200.0/24, ubest/mbest: 1/0
```

```
    *via 192.0.2.3%default, [20/0], 23:54:14, bgp-65421, external, tag 65500, segid: 10300 tunnelid:
0xc0000203 encap: VXLAN

10.100.200.3/32, ubest/mbest: 1/0
    *via 192.0.2.3%default, [20/0], 23:02:06, bgp-65421, external, tag 65500, segid: 10300 tunnelid:
0xc0000203 encap: VXLAN
```

A similar process happens for the 10.100.100.0/24 subnet behind leaf1, however, there is an important consideration on the Junos OS side – in Junos OS 22.2R2, a specific import policy is needed to insert the host route (/32 route) into the VRF table, even if the Type-2 MAC+IP route is correctly accepted into the EVPN table. The policy simply needs to match on the L2VNI route-target (the MAC VRF route-target) and accept it.

```
Policy-options {
    policy-statement ECMP {
        then {
            load-balance per-flow;
        }
    }
    policy-statement allow-loopback {
        from interface lo0.0;
        then accept;
    }
    policy-statement v100_accept {
        term 1 {
            from community v100;
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    community v100 members target:100:100;
}
```

This policy is then imported into the mac-vrf using '*vrf-import -import v100_accept*'. With this configuration in place, a host route for h1 should now be present in the VRF table on leaf3.

```
root@dc-tme-qfx5120-04> show route table Tenant1.inet.0

Tenant1.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.100.100.0/24    *[EVPN/170] 1d 09:04:41
                    >  to 198.51.100.5 via xe-0/0/0.0
10.100.200.0/24    *[Direct/0] 1d 09:04:40
                    >  via irb.200
10.100.200.3/32    *[EVPN/7] 1d 08:12:33
                    >  via irb.200
10.100.200.252/32  *[Local/0] 1d 09:04:40
                       Local via irb.200
10.100.200.254/32  *[Local/0] 1d 09:04:40
                       Local via irb.200
```

Host h3 should now be able to reach h1, which confirms that we have successfully built a fabric for Symmetric IRB interoperability between NXOS and Junos OS.

```
H3> ping
10.100.200.3
PING 10.100.200.3 (10.100.200.3): 56 data bytes
64 bytes from 10.100.200.3: icmp_seq=0 ttl=64 time=0.055 ms
64 bytes from 10.100.200.3: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 10.100.200.3: icmp_seq=2 ttl=64 time=0.099 ms
64 bytes from 10.100.200.3: icmp_seq=3 ttl=64 time=0.092 ms
^C
--- 10.100.200.3 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.040/0.072/0.099/0.025 ms
```

DCI between NXOS and Junos OS Fabrics for Bridged Overlay

Overview and Topology

The most common use case of interoperability is when two fabrics (or pods) comprising of different vendors need to talk to each other. It is less common to mix vendors in the same pod/fabric.

To demonstrate such a use case, we will use the following topology:

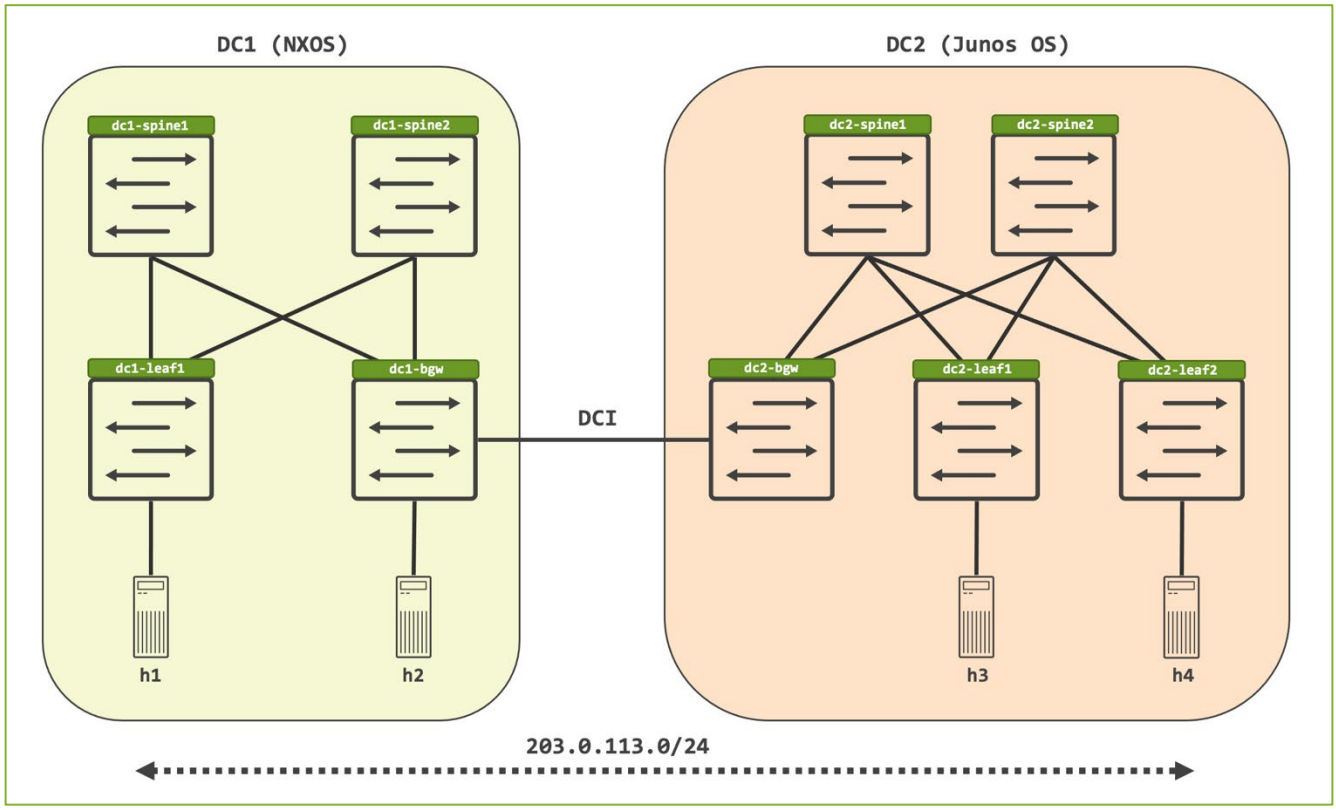


Figure 9: Topology of Two Vendor Fabrics Connected Via DCI

As seen above, DC1 comprises of a NXOS 3-stage Clos architecture with leaf2 (named dc1-bgw) acting as a border leaf. This border leaf connects to DC2, which is a Junos OS based 3-stage Clos architecture. DC2 has a dedicated border leaf, dc2-bgw. All four hosts (h1 through h4) are in the same subnet – 203.0.113.0/24. The goal is to extend this bridged overlay across DCs and ensure that all hosts can reach other.

RFC 9014 and draft-sharma-bess-multi-site-evpn

NXOS does not implement RFC 9014 by default – NXOS implements IETF draft called draft-sharma-bess-multi-site-evpn. This is currently an active IETF draft. The NXOS implementation, as the draft name implies, is called EVPN multisite.

Junos OS and Junos OS Evolved implement RFC 9014. The Junos OS and Junos OS Evolved terminology is called VXLAN stitching. This paper does not aim to address the advantages of using VXLAN stitching or EVPN multisite over a traditional OTT (over the top) DCI. This document demonstrates how VXLAN stitching can interoperate with EVPN multisite, how both sides need to be configured for this to work, along with details of how this works.

Both EVPN multisite and RFC 9014 focus on DCI using gateways (GWs). EVPN accomplishes the need for segmentation of Data Centers into multiple domains/regions by using anycast GWs, using the same virtual IP address across multiple GWs per Data Center site. The reason for this choice was to reduce overhead of overlay ECMP.

Junos OS and Junos OS Evolved, as described in RFC 9014, uses Interconnect ESI (I-ESI) for high-availability of GWs. To support this a recursive resolution for the I-ESI must be performed with the use of EVPN Type-1 routes. I-ESI is not your traditional Ethernet Segment (ES) that is mapped against a physical link; it is a logical/virtual ES that allows for multiple GWs to exist per DC site.

Configuring EVPN Multisite on NXOS

The spine and leaf configurations are the same as the bridged overlay section. We'll post the configuration from one spine and one leaf.

A snippet from dc1-spine1.

```
Feature telnet
nv overlay evpn
feature bgp
feature vn-segment-vlan-based
feature lldp
feature nv overlay
!
route-map allow-loopback permit 10
  match interface loopback0
route-map nh-unchanged permit 10
  set ip next-hop unchanged
!
interface loopback0
  ip address 192.0.2.11/32
!
router bgp 65500
  router-id 192.0.2.11
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
  address-family l2vpn evpn
    retain route-target all
    allow-vni-in-ethertag
  template peer evpn
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
      route-map nh-unchanged out
  neighbor 192.0.2.1
    inherit peer evpn
    remote-as 65421
  neighbor 192.0.2.100
    inherit peer evpn
    remote-as 65425
  neighbor 198.51.100.0
    remote-as 65421
    address-family ipv4 unicast
  neighbor 198.51.100.16
    remote-as 65425
    address-family ipv4 unicast
```

A snippet from dc1-leaf1:

```
nv overlay evpn
feature bgp
feature vn-segment-vlan-based
feature lldp
feature nv overlay
!
vlan 100
  vn-segment 10100
!
route-map allow-loopback permit 10
  match interface loopback0
route-map permit-all permit 10
!
```

```
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback0
  member vni 10100
    ingress-replication protocol bgp
!
interface loopback0
  ip address 192.0.2.1/32
!
router bgp 65421
  router-id 192.0.2.1
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
    maximum-paths 4
  address-family l2vpn evpn
    advertise-pip
    allow-vni-in-ethertag
  template peer evpn
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 192.0.2.11
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.12
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.22
    inherit peer evpn
    remote-as 65500
  neighbor 198.51.100.1
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.3
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.9
    remote-as 65500
    address-family ipv4 unicast
  vrf Tenant1
    address-family ipv4 unicast
      redistribute direct route-map permit-all
evpn
  vni 10100 l2
    rd 192.0.2.1:100
    route-target import 1:1
    route-target export 1:1
```

As we see on dc1-leaf1, VNI 10100 is configured with a route-target of 1:1. This is important to note since route-target control becomes crucial for DCI (as we will see later).

On the NXOS border leaf (dc1-bgw), we configure a multisite site/domain ID. A Type-4 EVPN route is generated, with the site ID encoded in the ESI. This allows for DF/nDF election for BUM traffic forwarding. We also enable RFC 9014 interoperability mode using '*dci-advertise-pip*'. This now generates aType-1 route as well for the Ethernet Segment.

If multiple gateways exist at the same site, they must be configured with the same site/domain ID.

```
Nv overlay evpn
feature bgp
feature fabric forwarding
feature vn-segment-vlan-based
feature lldp
feature nv overlay
!
evpn multisite border-gateway 100
  dci-advertise-pip
```

```
!
vlan 100
  vn-segment 10100
```

The VXLAN interface (nve1, in this case) must specify a unique loopback to be used for the DCI (loopback100, in this case). We must also configure ingress replication explicitly for multisite since that is the only supported mode for replicating multi-destination traffic across the DCI.

```
Interface loopback0
  ip address 192.0.2.100/32
!
interface loopback100
  ip address 192.0.2.101/32
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback0
  multisite border-gateway interface loopback100
  member vni 10100
    multisite ingress-replication
    ingress-replication protocol bgp
```

In addition to this, the fabric facing interfaces and the DCI facing interfaces must be configured for tracking like below:

```
interface Ethernet1/50
  ip address 198.51.100.18/31
  no shutdown
  evpn multisite fabric-tracking
!
interface Ethernet1/51
  speed 40000
  no negotiate auto
  ip address 198.51.100.24/31
  no shutdown
  evpn multisite dci-tracking
!
interface Ethernet1/52
  mtu 9216
  ip address 198.51.100.16/31
  no shutdown
  evpn multisite fabric-tracking
!
```

From a BGP perspective, any DCI peer must be configured as ‘peer-type fabric-external’, and we also swap the route-target as the updates are sent out the DCI towards the remote GWs using an outbound route-map. We’ll understand why this is necessary shortly when we look at how the control plane EVPN updates are exchanged.

```
Route-map allow-loopback permit 10
  match interface loopback0
route-map allow-loopback permit 20
  match interface loopback100
route-map dci_rt permit 10
  set extcommunity rt 100:100
!
router bgp 65425
  router-id 192.0.2.100
  log-neighbor-changes
  address-family ipv4 unicast
    redistribute direct route-map allow-loopback
  address-family l2vpn evpn
    allow-vni-in-ethertag
  template peer evpn
    update-source loopback0
    ebgp-multihop 2
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 192.0.2.11
```

```
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.12
    inherit peer evpn
    remote-as 65500
  neighbor 192.0.2.200
    inherit peer evpn
    remote-as 65426
    peer-type fabric-external
    address-family l2vpn evpn
      route-map dci_rt out
  neighbor 198.51.100.17
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.19
    remote-as 65500
    address-family ipv4 unicast
  neighbor 198.51.100.25
    remote-as 65426
    address-family ipv4 unicast
```

Finally, we also explicitly import the DCI route-target, 100:100. Again, we'll understand why this is configured as we get to the control plane updates between sites.

```
Evpn
  vni 10100 l2
    rd 192.0.2.100:100
    route-target import 100:100
    route-target import 1:1
    route-target export 1:1
```

Configuring VXLAN Stitching on Junos OS/Junos OS Evolved

Like EVPN multisite, VXLAN stitching on Junos OS and Junos OS Evolved does not require any additional configuration on the spines and leafs themselves – this is the same as the bridged overlay configuration. Our focus is on DC2s border leaf, dc2-bgw.

To enable VXLAN stitching, the 'interconnect' hierarchy must be used within 'protocol evpn'. We'll also configure a mac-vrf routing-instance for this.

```
root@dc2-bgw# show routing-instances v100_macvrf
instance-type mac-vrf;
protocols {
    evpn {
        encapsulation vxlan;
        extended-vni-list 10100;
        interconnect {
            vrf-target target:100:100;
            route-distinguisher 192.0.2.200:200;
            esi {
                00:00:00:00:00:00:00:00:00:22;
                all-active;
            }
            interconnected-vni-list 10100;
            encapsulation vxlan;
        }
    }
}
vtep-source-interface lo0.0;
service-type vlan-based;
route-distinguisher 192.0.2.200:1;
vrf-target target:1:2;
vlans {
    v100 {
        vlan-id 100;
        vxlan {
            vni 10100;
        }
    }
}
```

As seen in the above configuration, the ‘interconnect’ option requires the following key components:

- A unique interconnect route-target
- A unique interconnect route-distinguisher
- An ESI (called the I-ESI)

In our case, we use an interconnect route-target of 100:100, an interconnect route-distinguisher of 192.0.2.200:200, and an I-ESI of 00:00:00:00:00:00:00:00:00:22.

EVPN must also be configured with both the ‘extended-vni-list’ which describes the VNIs enabled locally and an ‘interconnected-vni-list’ which describes the VNIs enabled/extended across the DCI. Through this logical and hierarchical configuration, you have granular control over what VNIs are extended over the DCI.

Putting it all Together – Understanding How Updates are Exchanged Over the DCI

As this use case is bridged overlay only, the focus is on EVPN Type-2 routes, and resolving the remaining issues is simple: the GWs (for both EVPN multisite and VXLAN stitching) re-originate locally learnt routes into the DCI by inserting themselves as the next-hop in the EVPN update.

This allows for a clear demarcation of tunnels between sites: local leafs form VXLAN tunnels with the local GW, the local GW forms a VXLAN tunnel with the remote GW and the remote GW forms a VXLAN tunnel with the remote leafs. Visually, this can be represented as below. For simplicity, the tunnels are seen as traversing only via one spine, but since this is ECMP, a flow could go through any spine.

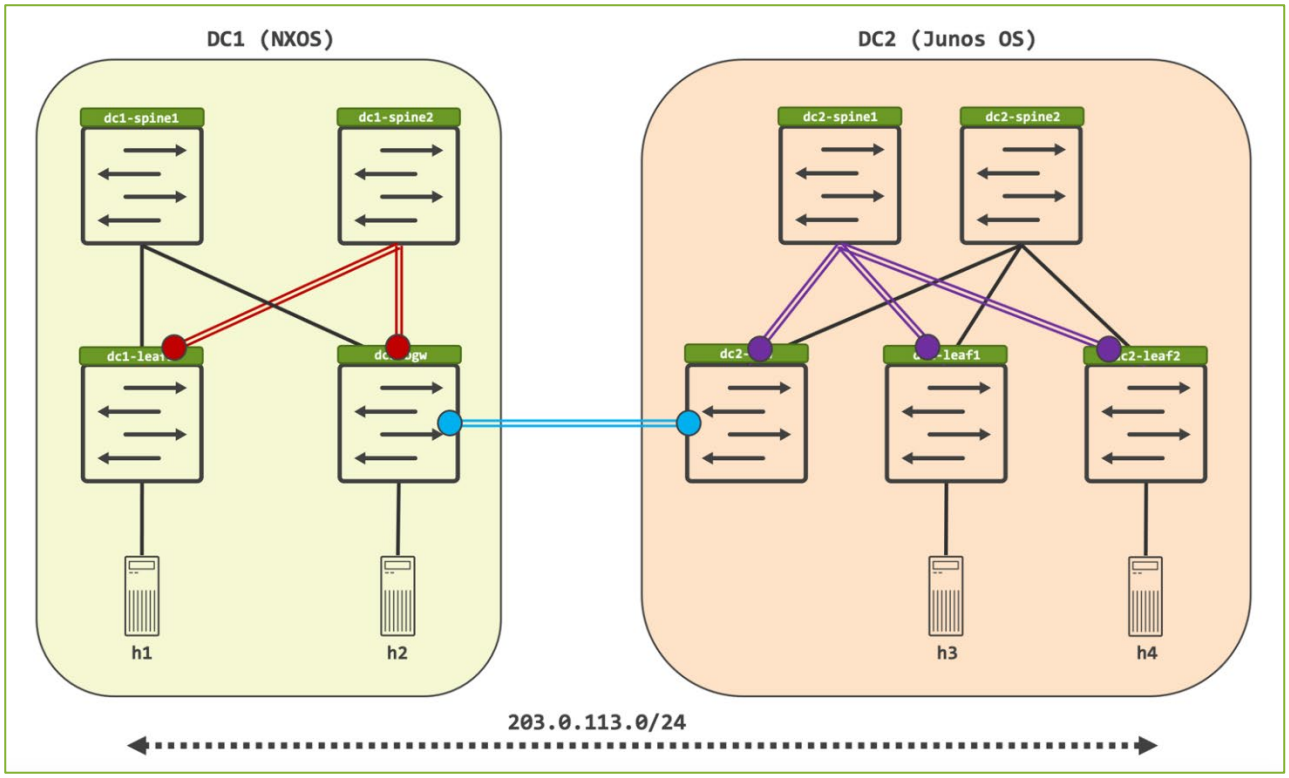


Figure 10: DCI Interconnection Between Pods from Two Different Vendors

With VXLAN stitching on Junos OS and Junos OS Evolved, as the update is sent out the DCI, several attributes are changed for the re-originated route:

- The next hop is set to self.
- The route-target is changed to the interconnect route-target.
- The route-distinguisher is set to the interconnect route-distinguisher.
- The I-ESI is added to the route.

Visually, this is what it looks like:

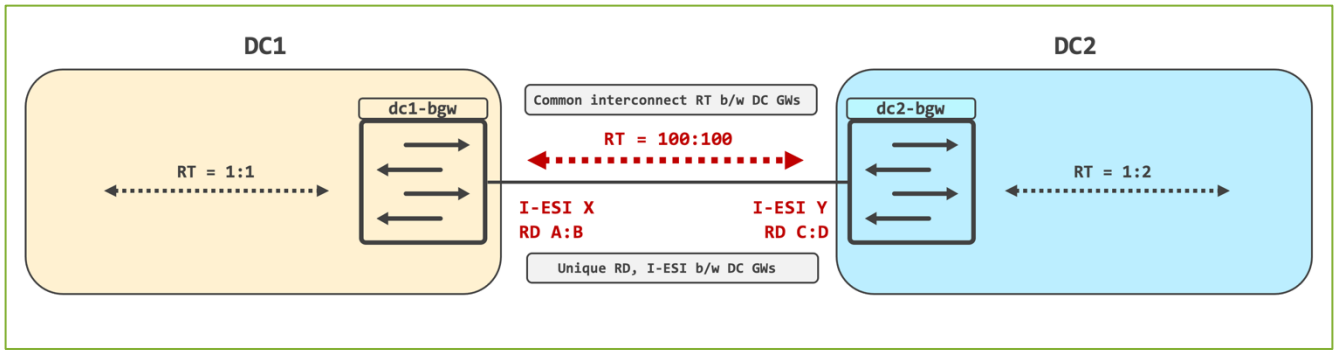


Figure 11: Route-Targets Across DCs

Following Control Plane EVPN Updates from DC2 to DC1

Let's trace the flow of h3's MAC address from DC2 to DC1 from a control plane EVPN update perspective. First, on dc2-leaf1, h3's MAC address is a local learn.

```
root@dc2-leaf1 # run show ethernet-switching table f0:4b:3a:b9:81:13

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 7 entries, 5 learned
Routing instance : v100_macvrf
   Vlan                MAC                 MAC      Logical                 SVLBNH/      Active
   name                address             flags    interface               VENH Index   source
   v100                f0:4b:3a:b9:81:13   D        et-0/0/54.0
```

This is sent as a BGP EVPN update to the spines. Let's take dc2-spine1 as an example, and confirm that it has received this route from dc2-leaf.

```
root@dc2-spine1# run show route receive-protocol bgp 192.0.2.3 evpn-mac-address f0:4b:3a:b9:81:13 table
bgp.evpn.0

bgp.evpn.0: 25 destinations, 25 routes (25 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:192.0.2.3:1::0::f0:4b:3a:b9:81:13/304 MAC/IP
*                         192.0.2.3                               65423 I
  2:192.0.2.3:1::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP
*                         192.0.2.3                               65423 I
```

More importantly, the route-target is 1:2, which is the local route-target and the route-distinguisher is of dc2-leaf1, as expected.

```
root@dc2-spine1# run show route receive-protocol bgp 192.0.2.3 evpn-mac-address f0:4b:3a:b9:81:13 table
bgp.evpn.0 extensive

bgp.evpn.0: 25 destinations, 25 routes (25 active, 0 holddown, 0 hidden)
* 2:192.0.2.3:1::0::f0:4b:3a:b9:81:13/304 MAC/IP (1 entry, 1 announced)
     Accepted
     Route Distinguisher: 192.0.2.3:1
     Route Label: 10100
```

```
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 192.0.2.3
     AS path: 65423 I
     Communities: target:1:2 encapsulation:vxlan(0x8)

* 2:192.0.2.3:1::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP (1 entry, 1 announced)
     Accepted
     Route Distinguisher: 192.0.2.3:1
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 192.0.2.3
     AS path: 65423 I
     Communities: target:1:2 encapsulation:vxlan(0x8)
```

The spines will send this to the other leaf switches, including dc2-bgw. There should be no change in any of these attributes, including next hop as the spine sends this out. We can confirm this on dc2-bgw.

```
root@dc2-bgw# run show route receive-protocol bgp 192.0.2.13 evpn-mac-address f0:4b:3a:b9:81:13 table
bgp.evpn.0 extensive

bgp.evpn.0: 25 destinations, 31 routes (25 active, 0 holddown, 0 hidden)
  2:192.0.2.3:1::0::f0:4b:3a:b9:81:13/304 MAC/IP (2 entries, 1 announced)
     Import Accepted
     Route Distinguisher: 192.0.2.3:1
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 192.0.2.3
     AS path: 65501 65423 I
     Communities: target:1:2 encapsulation:vxlan(0x8)

  2:192.0.2.3:1::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP (2 entries, 1 announced)
     Import Accepted
     Route Distinguisher: 192.0.2.3:1
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 192.0.2.3
     AS path: 65501 65423 I
     Communities: target:1:2 encapsulation:vxlan(0x8)
```

Below we can see border leaf dc2-bgw send this update to dc1-bgw.

```
root@dc2-bgw# run show route advertising-protocol bgp 192.0.2.100 evpn-mac-address f0:4b:3a:b9:81:13
table bgp.evpn.0

bgp.evpn.0: 25 destinations, 31 routes (25 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:192.0.2.3:1::0::f0:4b:3a:b9:81:13/304 MAC/IP
*                         Self                                    65501 65423 I
  2:192.0.2.200:200::0::f0:4b:3a:b9:81:13/304 MAC/IP
*                         Self                                    I
  2:192.0.2.3:1::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP
*                         Self                                    65501 65423 I
  2:192.0.2.200:200::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP
*                         Self                                    I
```

Looking at this with the ‘extensive’ keyword, we see that the attributes have now changed. The update is sent out with the interconnect route-distinguisher, route-target and the I-ESI attached to the route.

```
root@dc2-bgw# run show route advertising-protocol bgp 192.0.2.100 evpn-mac-address f0:4b:3a:b9:81:13
table bgp.evpn.0 extensive

Warning: License key missing; requires 'bgp' license


bgp.evpn.0: 25 destinations, 31 routes (25 active, 0 holddown, 0 hidden)
* 2:192.0.2.3:1::0::f0:4b:3a:b9:81:13/304 MAC/IP (2 entries, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 192.0.2.3:1
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: Self
     AS path: [65426] 65501 65423 I
```

```
     Communities: target:1:2 encapsulation:vxlan(0x8)

* 2:192.0.2.200:200::0::f0:4b:3a:b9:81:13/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 192.0.2.200:200
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:22
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65426] I
     Communities: target:100:100 encapsulation:vxlan(0x8)

* 2:192.0.2.3:1::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP (2 entries, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 192.0.2.3:1
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: Self
     AS path: [65426] 65501 65423 I
     Communities: target:1:2 encapsulation:vxlan(0x8)

* 2:192.0.2.200:200::0::f0:4b:3a:b9:81:13::203.0.113.3/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 192.0.2.200:200
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:22
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65426] I
     Communities: target:100:100 encapsulation:vxlan(0x8)
```

Because of this, it is necessary to have an import statement for this route-target on dc1-gw.

```
evpn
  vni 10100 l2
    rd 192.0.2.100:100
    route-target import 100:100
    route-target import 1:1
    route-target export 1:1
```

This route is now accepted into BGP RIB on dc1-gw.

```
dc1-bgw# show bgp l2vpn evpn f0:4b:3a:b9:81:13
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.100:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[0]:[0.0.0.0]/216, version 462
Paths: (1 available, best #1)
Flags: (0x000212) (high32 0x000400) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.200:200:[2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[0]:[0.0.0.0]/216
  AS-Path: 65426 , path sourced external to AS
    192.0.2.200 (metric 0) from 192.0.2.200 (192.0.2.200)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0022

  Path-id 1 (dual) advertised to peers:
    192.0.2.11         192.0.2.12
BGP routing table entry for [2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[32]:[203.0.113.3]/248, version 464
Paths: (1 available, best #1)
Flags: (0x000212) (high32 0x000400) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.200:200:[2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[32]:[203.0.113.3]/248
  AS-Path: 65426 , path sourced external to AS
    192.0.2.200 (metric 0) from 192.0.2.200 (192.0.2.200)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
```

```
      ESI: 0000.0000.0000.0000.0022

  Path-id 1 (dual) advertised to peers:
    192.0.2.11         192.0.2.12

Route Distinguisher: 192.0.2.200:200
BGP routing table entry for [2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[0]:[0.0.0.0]/216, version 457
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: L2-10100
  AS-Path: 65426 , path sourced external to AS
    192.0.2.200 (metric 0) from 192.0.2.200 (192.0.2.200)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0022

  Path-id 1 not advertised to any peer
BGP routing table entry for [2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[32]:[203.0.113.3]/248, version 459
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: L2-10100
  AS-Path: 65426 , path sourced external to AS
    192.0.2.200 (metric 0) from 192.0.2.200 (192.0.2.200)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:100:100 ENCAP:8
      ESI: 0000.0000.0000.0000.0022

  Path-id 1 not advertised to any peer
```

It is installed in l2route and eventually the MAC address table, since the recursive lookup for the ESI is successful (if a Type-1 route is present for the ESI):

```
dc1-bgw# show l2route evpn mac all detail

Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv (AD):Auto-Delete (D):Del Pending
(S):Stale (C):Clear, (Ps):Peer Sync (O):Re-Originated (Nho):NH-Override
(Asy):Asymmetric (Gw):Gateway
(Pf):Permanently-Frozen, (Orp): Orphan

(PipOrp): Directly connected Orphan to PIP based vPC BGW
(PipPeerOrp): Orphan connected to peer of PIP based vPC BGW
Topology    Mac Address    Prod   Flags         Seq No     Next-Hops
----------- -------------- ------ ------------- ---------- --------------------------------------

*snip*

100         f04b.3ab9.8113 BGP    SplRcv        0          192.0.2.200 (Label: 10100)
            Route Resolution Type: ESI
            Forwarding State: Resolved (PL)
            Resultant PL: 192.0.2.200
            Sent To: L2FM
            ESI : 0000.0000.0000.0000.0022
            Encap: 1

*snip*

dc1-bgw# show l2route evpn ead all detail

Flags -(A):Active (S):Standby (V):Virtual ESI (D):Del Pending (S):Stale

Topology ID   Prod   ESI                       NFN Bitmap  Num PLs Flags
------------ ------ ------------------------- ------------ ------ ------
100           BGP    0000.0000.0000.0000.0022   0          1      -
```

```
               Next-Hops: 192.0.2.200
4294967294    BGP    0000.0000.0000.0000.0022   0          1      A
               Next-Hops: 192.0.2.200
4294967294    VXLAN  0300.0000.0000.6400.0309   128        0      AV
               Next-Hops: 192.0.2.100

tme-nexus03# show mac address-table address f04b.3ab9.8113
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
        (NA)- Not Applicable
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C  100     f04b.3ab9.8113   dynamic  NA         F      F    nve1(192.0.2.200)
```

This is sent as a BGP EVPN update to the spines, and in turn, dc1-leaf1. While advertising remote routes, GW is expected to re-originate the routes using its own route-distinguisher, route-target, and local I-ES.

On dc1-spine1, for example, the route-distinguisher is 100:10100. This is the GW's automatically generated route-distinguisher for route re-origination. The route-target is 1:1.

Notice, however, that the I-ESI is missing. This is because for the local DC, EVPN multisite advertises the routes using the anycast IP address (192.0.2.101, in this case), and thus, there is no requirement of the I-ESI to be attached to the route.

```
dc1-spine1# show bgp l2vpn evpn f0:4b:3a:b9:81:13
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 100:10100
BGP routing table entry for [2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[0]:[0.0.0.0]/216, version 185
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
  AS-Path: 65425 65426 , path sourced external to AS
    192.0.2.101 (metric 0) from 192.0.2.100 (192.0.2.100)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:1:1 ENCAP:8

  Path-id 1 advertised to peers:
    192.0.2.1
BGP routing table entry for [2]:[0]:[0]:[48]:[f04b.3ab9.8113]:[32]:[203.0.113.3]/248, version 187
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
  AS-Path: 65425 65426 , path sourced external to AS
    192.0.2.101 (metric 0) from 192.0.2.100 (192.0.2.100)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:1:1 ENCAP:8

  Path-id 1 advertised to peers:
    192.0.2.1
```

The update is reflected to the leaf, and dc1-leaf1 now installs this in the MAC address table with a next hop of the anycast gateway IP address:

```
dc1-leaf1# show mac address-table address f0:4b:3a:b9:81:13
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
        (NA)- Not Applicable
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C  100     f04b.3ab9.8113   dynamic  NA         F      F    nve1(192.0.2.101)
```

This concludes the control plane updates from DC2 to DC1.

Following Control Plane EVPN Updates from DC1 to DC2

On dc1-leaf1, h1's MAC address is a local learn.

```
dc1-leaf1# show mac address-table address 0010.9400.0003
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan,
        (NA)- Not Applicable
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  100     0010.9400.0003   dynamic  NA         F      F    Eth1/10
```

It is inserted into BGP RIB and sent out as an update to the spines.

```
dc1-leaf1# show bgp l2vpn evpn 0010.9400.0003
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[0010.9400.0003]:[0]:[0.0.0.0]/216, version 679
Paths: (1 available, best #1)
Flags: (0x000102) (high32 00000000) on xmit-list, is not in l2rib/evpn

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    192.0.2.1 (metric 0) from 0.0.0.0 (192.0.2.1)
      Origin IGP, MED not set, localpref 100, weight 32768
      Received label 10100
      Extcommunity: RT:1:1 ENCAP:8

  Path-id 1 advertised to peers:
    192.0.2.11         192.0.2.12
```

The route-target and route-distinguisher used are the ones configured for the VNI.

```
evpn
  vni 10100 l2
    rd 192.0.2.1:100
    route-target import 1:1
    route-target export 1:1
```

The spines will reflect this to dc1-bgw. As you can see, the next hop remains dc1-leaf1 since the spines are configured not to modify the next hop for EVPN updates.

```
dc1-bgw# show bgp l2vpn evpn 0010.9400.0003
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.0.2.1:100
BGP routing table entry for [2]:[0]:[0]:[48]:[0010.9400.0003]:[0]:[0.0.0.0]/216, version 415
Paths: (2 available, best #2)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

  Path type: external, path is valid, not best reason: newer EBGP path, no labeled nexthop
  AS-Path: 65500 65421 , path sourced external to AS
```

```
    192.0.2.1 (metric 0) from 192.0.2.12 (192.0.2.12)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:1:1 ENCAP:8

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: L2-10100
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:1:1 ENCAP:8

  Path-id 1 not advertised to any peer

Route Distinguisher: 192.0.2.100:100    (L2VNI 10100)
BGP routing table entry for [2]:[0]:[0]:[48]:[0010.9400.0003]:[0]:[0.0.0.0]/216, version 451
Paths: (1 available, best #1)
Flags: (0x000212) (high32 0x000400) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.0.2.1:100:[2]:[0]:[0]:[48]:[0010.9400.0003]:[0]:[0.0.0.0]/216
  AS-Path: 65500 65421 , path sourced external to AS
    192.0.2.1 (metric 0) from 192.0.2.11 (192.0.2.11)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10100
      Extcommunity: RT:1:1 ENCAP:8

  Path-id 1 (dual) advertised to peers:
    192.0.2.200
```

The border leaf, dc1-bgw, should now send this update across the DCI to dc2-bgw. This is where our route-map overrides the route-target for updates going to dc2-bgw. Since NXOS EVPN multisite does not include any clear demarcation of configuration for the interconnect, this route-map is necessary for interoperability.

The route-map overrides the route-target to 100:100, which is the interconnect route-target on dc2-bgw, and it is applied outbound for all DCI neighbors:

```
route-map dci_rt permit 10
  set extcommunity rt 100:100

router bgp 65425
  router-id 192.0.2.100
  log-neighbor-changes

*snip*

  neighbor 192.0.2.200
    inherit peer evpn
    remote-as 65426
    peer-type fabric-external
    address-family l2vpn evpn
      route-map dci_rt out

*snip*
```

On dc2-bgw, if we look at the received route, we see this new route-target as expected. The route-distinguisher is also changed, the I-ESI is attached, and the next hop is changed to self.

```
root@dc2-bgw# run show route receive-protocol bgp 192.0.2.100 evpn-mac-address 0010.9400.0003 table
bgp.evpn.0 extensive

Warning: License key missing; requires 'bgp' license


bgp.evpn.0: 25 destinations, 31 routes (25 active, 0 holddown, 0 hidden)
* 2:100:10100::0::00:10:94:00:00:03/304 MAC/IP (1 entry, 1 announced)
     Import Accepted
     Route Distinguisher: 100:10100
```

```
     Route Label: 10100
     ESI: 03:00:00:00:00:00:64:00:03:09
     Nexthop: 192.0.2.100
     MED: 2000
     AS path: 65425 65500 65421 I
     Communities: target:100:100 encapsulation:vxlan(0x8)
```

When dc2-bgw sends this out towards the spines, it needs to re-originate the route with the local route-distinguisher (not the interconnect route-distinguisher), the local route-target (not the interconnect route-target), the I-ESI attached to the route, and the next hop changed to self.

```
root@dc2-bgw# run show route advertising-protocol bgp 192.0.2.13 evpn-mac-address 0010.9400.0003 table
bgp.evpn.0 extensive

Warning: License key missing; requires 'bgp' license


bgp.evpn.0: 25 destinations, 31 routes (25 active, 0 holddown, 0 hidden)
* 2:192.0.2.200:1::0::00:10:94:00:00:03/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 192.0.2.200:1
     Route Label: 10100
     ESI: 00:00:00:00:00:00:00:00:00:22
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65426] I
     Communities: target:1:2 encapsulation:vxlan(0x8)
```

The spines will reflect this to dc2-leaf1, and it will accept this route since the vrf-target matches the route-target in the route and the ESI can be resolved successfully. Finally, this will be pulled into the EVPN database and installed in the ethernet switching-table on dc2-leaf1.

```
root@dc2-leaf1# show routing-instances v100_macvrf
instance-type mac-vrf;
protocols {
    evpn {
        encapsulation vxlan;
        extended-vni-list 10100;
    }
}
vtep-source-interface lo0.0;
service-type vlan-based;
interface et-0/0/54.0;
route-distinguisher 192.0.2.3:1;
vrf-target target:1:2;
vlans {
    v100 {
        vlan-id 100;
        vxlan {
            vni 10100;
        }
    }
}

root@dc2-leaf1# run show evpn database mac-address 0010.9400.0003 extensive
Instance: v100_macvrf

VN Identifier: 10100, MAC address: 00:10:94:00:00:03
  State: 0x0
  Source: 00:00:00:00:00:00:00:00:00:22, Rank: 1, Status: Active
    Remote origin: 192.0.2.200
    Remote state: <Mac-Only-Adv>
    Mobility sequence number: 0 (minimum origin address 192.0.2.200)
    Timestamp: Feb 05 19:10:01.919062 (0x63e06f89)
    State: <Remote-To-Local-Adv-Done>
    MAC advertisement route status: Not created (no local state present)
    History db: <No entries>

root@dc2-leaf1# run show ethernet-switching table 0010.9400.0003

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)
```

```
Ethernet switching table : 8 entries, 5 learned
Routing instance : v100_macvrf
   Vlan                MAC                 MAC      Logical                SVLBNH/      Active
   name                address             flags    interface              VENH Index   source
   v100                00:10:94:00:00:03   DR       esi.1745               1743
00:00:00:00:00:00:00:00:00:22
```

This concludes the flow of control plane EVPN updates from DC1 to DC2.