EVPN VXLAN Interoperability Between Arista EOS and Junos OS

Author: Shalini Mukherjee

Published: 2/18/2025

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS (Operating System) has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

Table of Contents

Executive Summary

This document is a technical exploration of interoperability issues between Arista's EOS 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 Arista's EOS and switches running Juniper's Junos OS and Junos OS Evolved. 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. Arista 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 Arista's EOS and Juniper's Junos OS and Junos OS Evolved 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 explores 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 Arista's EOS MLAG and Junos OS.
- A bridged overlay fabric with Arista's EOS ESI-LAG and Junos OS ESI-LAG.
- Asymmetric IRB between Arista's EOS and Junos OS.
- Symmetric IRB between Arista's EOS and Junos OS.
- DCI between an Arista's EOS fabric and a Junos OS fabric for a bridged overlay.
- DCI between Arista's EOS and Junos OS Fabrics using Type-5 stitching

Arista's EOS devices are used as fabric spines throughout the document.
This whitepaper describes the configuration for these use cases to work, provides configuration examples. Junos OS Evolved is used as Border leaf switches in the DCI sections, but only Junos OS is used as Leaf switches.

Bridged Overlay with Arista's EOS MLAG and Junos OS

This case consists of:

- A data center network fabric with Standalone Arista EOS switch.
- Two Arista EOS switches forming a Multi Chassis Link Aggregation Lag (MLAG) pair.
- Two Arista EOS switches as all active multihoming pair by assigning an Ethernet Switch Identifier (ESI).
- Standalone Juniper Junos OS switch.
- Two Junos OS devices as Ethernet Switch Identifier (ESI) Link Aggregation Group (LAG) peers.

All pairs of switches are operating as leaf switches.

Multiple hosts are connected to the leaf switches:

- h1 is single homed to leaf1
- h2 is single homed to leaf2a, which is part of a MLAG pair
- h3 is connected to leaf2a and leaf2b via MLAG (the Arista pair)
- h4 is connected to leaf3a and leaf3b via ESI-LAG (using EVPN LAG multihoming) (Arista pair)
- h5 is connected to leaf5a and leaf5b via ESI-LAG (using EVPN LAG multihoming) (Juniper pair)
- h6 is single homed to leaf7

Overview and Topology

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

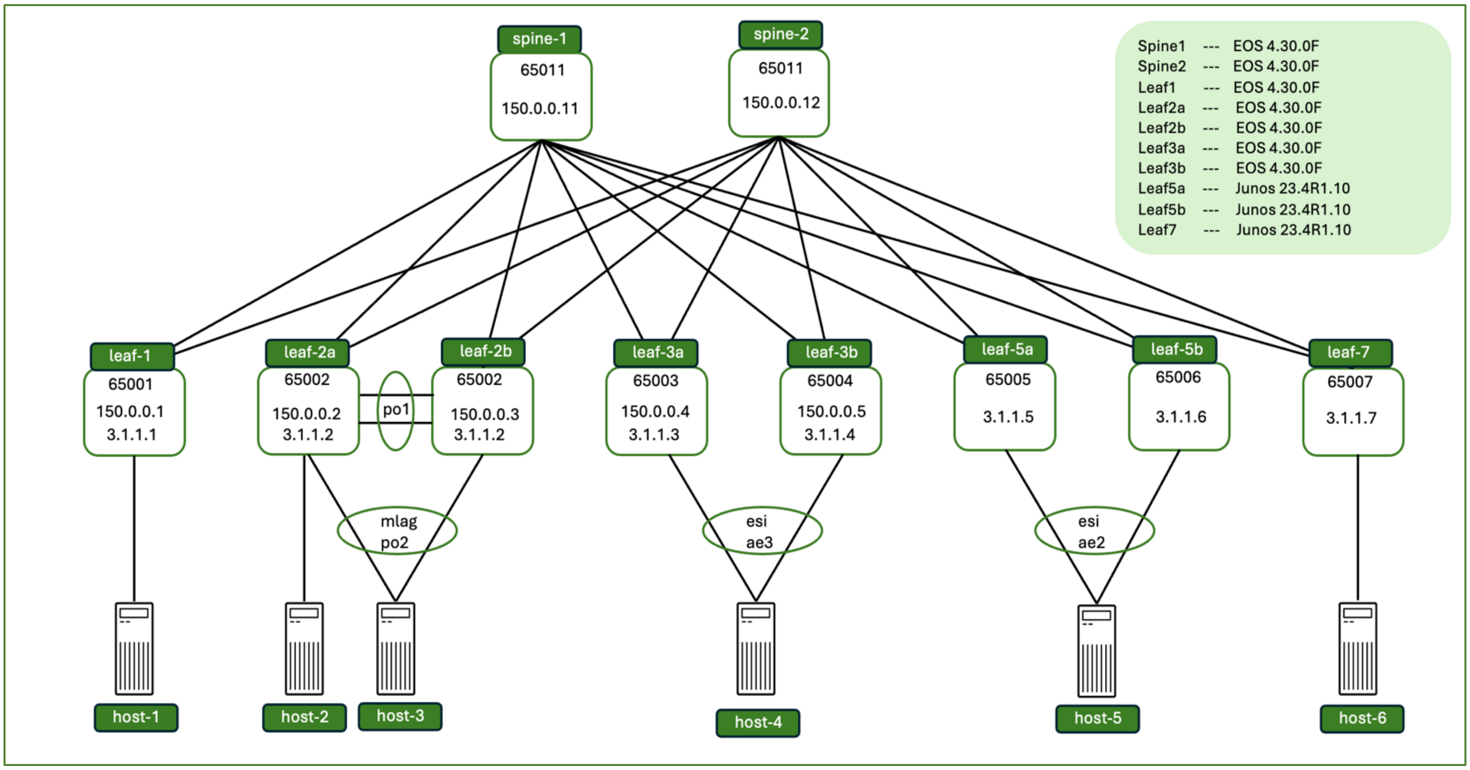

Figure 1 : Topology Diagram for Bridged Overlay

- The 3.1.1.0/24 range is used for loopback 0 of the leaf devices (Juniper and Arista leaf's)
- 150.0.0.0/24 range is used for loopback for the EOS devices (Arista leaf's and spines)
- The 11.0.0.0/24, 12.0.0.0/24 and 13.0.0.0/24 range is used for host addressing for VLAN 11,12 and 13 respectively with the last octet being the respective host number (1 for host-1, 2 for host-2 and so on).

Arista's EOS as Fabric Spines

Each Arista's EOS device is assigned a distinct IP address for its loopback 0 interface along with the configuration of IPv4 Underlay physical link. eBGP neighborship is formed by using the loopback 0 IP address.

```
Spine-1(config-if-Lo0)#show active
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.11/32
```

Loopback 1 interface is not needed on Spines as it is not acting as a VTEP.

Here is an example configuration of a physical interface of Spine1 connected to the leaf switches.

```
interface Ethernet1
   mtu 9214
   no switchport
   ip address 161.0.1.0/31
!
interface Ethernet2
   mtu 9214
   no switchport
   ip address 161.0.1.2/31
!
interface Ethernet3
   mtu 9214
   no switchport
   ip address 161.0.1.4/31
!
**snip**
```

To advertise the prefixes associated with loopback0, we use the below route-map and prefix-list command.

```
ip prefix-list PL-loopback-1 seq 10 permit 3.1.1.0/24 eq 32
ip prefix-list PL-loopback-evpn seq 20 permit 151.0.0.0/24 eq 32
!
route-map RM-BGP permit 10
   match ip address prefix-list PL-loopback-evpn
!
route-map RM-BGP-1 permit 10
   match ip address prefix-list PL-loopback-1

router bgp 65011
*snip*
   redistribute connected route-map RM-BGP
```

The above prefix-list command will only match /32 prefixes for 3.1.1.0/24 and 151.0.0.0/24.

eBGP 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. A single BGP AS is used for underlay and overlay of a EVPN VXLAN fabric.

Below is an example from Spine1 (Arista's EOS) BGP configuration including underlay (IPv4-UNDERLAY) and overlay (EVPN-OVERLAY) peering.

```
Spine-1(config)#show running-configuration section router
router bgp 65011
   router-id 151.0.0.11
   no bgp default ipv4-unicast
   distance bgp 20 200 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY next-hop-unchanged
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor IPv4-UNDERLAY peer group
   neighbor IPv4-UNDERLAY send-community
   neighbor IPv4-UNDERLAY maximum-routes 9000
   neighbor 3.1.1.5 peer group EVPN-OVERLAY
   neighbor 3.1.1.5 remote-as 65005
   neighbor 3.1.1.5 description Leaf-5a
   neighbor 3.1.1.6 peer group EVPN-OVERLAY
   neighbor 3.1.1.6 remote-as 65006
   neighbor 3.1.1.6 description Leaf-5b
   neighbor 3.1.1.7 peer group EVPN-OVERLAY
   neighbor 3.1.1.7 remote-as 65007
   neighbor 3.1.1.7 description Leaf-7
   neighbor 151.0.0.1 peer group EVPN-OVERLAY
   neighbor 151.0.0.1 remote-as 65001
   neighbor 151.0.0.1 description Leaf-1
   neighbor 151.0.0.2 peer group EVPN-OVERLAY
   neighbor 151.0.0.2 remote-as 65002
```

```
   neighbor 151.0.0.2 description Leaf-2a
   neighbor 151.0.0.3 peer group EVPN-OVERLAY
   neighbor 151.0.0.3 remote-as 65002
   neighbor 151.0.0.3 description Leaf-2b
   neighbor 151.0.0.4 peer group EVPN-OVERLAY
   neighbor 151.0.0.4 remote-as 65003
   neighbor 151.0.0.4 description Leaf-3a
   neighbor 151.0.0.5 peer group EVPN-OVERLAY
   neighbor 151.0.0.5 remote-as 65004
   neighbor 151.0.0.5 description Leaf-3b
   neighbor 161.0.1.1 peer group IPv4-UNDERLAY
   neighbor 161.0.1.1 remote-as 65003
   neighbor 161.0.1.1 description Leaf-3a
   neighbor 161.0.1.3 peer group IPv4-UNDERLAY
   neighbor 161.0.1.3 remote-as 65004
   neighbor 161.0.1.3 description Leaf-3b
   neighbor 161.0.1.5 peer group IPv4-UNDERLAY
   neighbor 161.0.1.5 remote-as 65005
   neighbor 161.0.1.5 description Leaf-5a
   neighbor 161.0.1.7 peer group IPv4-UNDERLAY
   neighbor 161.0.1.7 remote-as 65006
   neighbor 161.0.1.7 description Leaf-5b
   neighbor 161.0.1.19 peer group IPv4-UNDERLAY
   neighbor 161.0.1.19 remote-as 65007
   neighbor 161.0.1.19 description Leaf-7
   neighbor 161.0.1.21 peer group IPv4-UNDERLAY
   neighbor 161.0.1.21 remote-as 65001
   neighbor 161.0.1.21 description Leaf-1
   neighbor 161.0.1.23 peer group IPv4-UNDERLAY
   neighbor 161.0.1.23 remote-as 65002
   neighbor 161.0.1.23 description Leaf-2a
   neighbor 161.0.1.25 peer group IPv4-UNDERLAY
   neighbor 161.0.1.25 remote-as 65002
   neighbor 161.0.1.25 description Leaf-2b
   redistribute connected route-map RM-BGP
   !
   address-family evpn
      bgp additional-paths send ecmp
      neighbor EVPN-OVERLAY activate
   !
   address-family ipv4
      no neighbor EVPN-OVERLAY activate
      neighbor IPv4-UNDERLAY activate
```

Considering Spine-1 as an example, BGP peering is established for both underlay and overlay. The AFI/SAFI column in the below example lists L2VPN EVPN for indicating overlay peering and IPv4 Unicast for underlay peering.

```
Spine-1(config)#show bgp summary
BGP summary information for VRF default
Router identifier 151.0.0.11, local AS number 65011
Neighbor            AS Session State AFI/SAFI                AFI/SAFI State   NLRI Rcd   NLRI Acc
---------- ----------- ------------- ----------------------- -------------- ---------- ----------
3.1.1.5          65005 Established   L2VPN EVPN              Negotiated              9          9
3.1.1.6          65006 Established   L2VPN EVPN              Negotiated              9          9
3.1.1.7          65007 Established   L2VPN EVPN              Negotiated              9          9
151.0.0.1        65001 Established   L2VPN EVPN              Negotiated              6          6
151.0.0.2        65002 Established   L2VPN EVPN              Negotiated              6          6
151.0.0.3        65002 Established   L2VPN EVPN              Negotiated              6          6
151.0.0.4        65003 Established   L2VPN EVPN              Negotiated              8          8
151.0.0.5        65004 Established   L2VPN EVPN              Negotiated             10         10
161.0.1.1        65003 Established   IPv4 Unicast            Negotiated              2          2
161.0.1.3        65004 Established   IPv4 Unicast            Negotiated              2          2
161.0.1.5        65005 Established   IPv4 Unicast            Negotiated              1          1
161.0.1.7        65006 Established   IPv4 Unicast            Negotiated              1          1
161.0.1.19       65007 Established   IPv4 Unicast            Negotiated              1          1
161.0.1.21       65001 Established   IPv4 Unicast            Negotiated              2          2
161.0.1.23       65002 Established   IPv4 Unicast            Negotiated              3          3
161.0.1.25       65002 Established   IPv4 Unicast            Negotiated              3          3
*snip*
```

Deploying a Bridged Overlay Fabric with Arista's EOS and Junos OS

- Leaf1 is a standalone Arista's EOS switch with Host1.
- Leaf2a and 2b are Arista's EOS switches in a MLAG pair with 2 hosts connected to them. Host2 is an orphan port connected to Leaf2a as a standalone whereas Host3 is multihomed to both via portchannel po2.
- Leaf3a and 3b are Arista's EOS switches in an ESI lag with Host4 connected via po3.
- Leaf5a and 5b are Junos OS switches also connected in an ESI lag with Host5 via ae2.
- Leaf7 is a standalone Junos OS switch connected to Host6.

All Arista's EOS leaf switches in the network will have loopback 0 interface for overlay peering and loopback 1 interface which will be the source for the VXLAN tunnel.

MLAG Implementation

The below configuration example is from the MLAG pair Leaf2a and2b depicting the steps needed to form the MLAG.

In MLAG deployments, loopback 1 IP address is shared between MLAG leaf switches to function as a single logical VTEP.
Below is the config of loopback0 and loopback1 on Leaf2a and Leaf2b. As you can see loopback0 IP address is unique on both mlag leafs since they have separate EBGP overlay peering with the Spines but loopback1 IP address is same to represent a single logical VTEP.

```
Leaf-2a(config)#show run interfaces lo0-1
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.2/32
interface Loopback1
   description VTEP_vxlan
   ip address 3.1.1.2/32
```

```
Leaf-2b(config)#show run interfaces lo0-1
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.3/32
interface Loopback1
   description VTEP_vxlan
   ip address 3.1.1.2/32
```

iBGP sessions are established between each MLAG leaf pair to address specific failure scenarios. These sessions enable traffic to traverse the peer-link between the MLAG peers in the event of a failure where a leaf switch loses its connections to the spine switches. A separate peer-group should be used for this MLAG iBGP peering, which can also be reused for any other MLAG domains in the network. A separate SVI (vlan4093) is also required to form the iBGP peering between the 2 MLAG peers.

```
Leaf-2a(config)#show running-configuration section bgp
*snip*
router bgp 65002
  neighbor mlag-peer-underlay peer group
   neighbor mlag-peer-underlay remote-as 65002
   neighbor mlag-peer-underlay next-hop-self
   neighbor mlag-peer-underlay description Leaf-2b
   neighbor mlag-peer-underlay route-map RM-mlag in
   neighbor mlag-peer-underlay send-community
   neighbor mlag-peer-underlay maximum-routes 0
   neighbor 9.2.5.1 peer group mlag-peer-underlay

   address-family ipv4
*snip*
      neighbor mlag-peer-underlay activate
```

```
Leaf-2a(config)#show run interface Vlan4093
interface Vlan4093
   description mlag-L3-peering
   mtu 9214
   ip address 9.2.5.0/31
```

Another SVI (vlan4094) is needed for communications between the MLAG peers. A physical portchannel LAG interface po1 configured as a trunk port will be needed between the peers. Both vlans 4093 and 4094 should be part of this portchannel po1.

```
Leaf-2a(config)#show running-configuration section vlan
*snip*
vlan 4093
   trunk group L3-peer
!
vlan 4094
   trunk group mlag
```

```
Leaf-2a(config)#show run interfaces port-Channel 1
interface Port-Channel1
   switchport mode trunk
   switchport trunk group L3-peer
   switchport trunk group mlag
```

```
Leaf-2a(config)#show run interfaces vlan4094
interface Vlan4094
   no autostate
   ip address 10.0.0.0/31
```

SVI 4094 is used under the MLAG config to establish peer communication via peer-link portchannel po1 as shown below

```
Leaf-2a(config)#show running-configuration section mlag
*snip*
mlag configuration
   domain-id 1
   local-interface Vlan4094
   peer-address 10.0.0.1
   peer-link Port-Channel1
```

To implement mlag in an EVPN VXLAN fabric, Arista's EOS switch needs **interface Vxlan1** which is a virtual interface that represents the VXLAN overlay network. It is used to establish and manage VXLAN tunnels for Layer 2 and Layer 3 connectivity across the fabric.
The key functions of vxlan1 interface includes encapsulation and decapsulation, mapping VLANs to VXLAN VNI's, integration with BGP EVPN to exchange VXLAN control plane information and acting as tunnel endpoints for VXLAN packets.

The loopback interface0 is used as a VTEP source interface for the orphan port where Host2 is connected or any other Host which is single homed to one of the MLAG pair. Whereas loopback interface1 is used for Host3 or for any hosts dual-homed to both MLAG pairs.

The **MLAG System-ID** under interface vxlan1 is a unique identifier used to represent the MLAG pair as a single logical entity. It is typically an MLAG-specific MAC address configured on both switches in the MLAG pair. This identifier ensures that the MLAG peers appear as a single device to the connected downstream devices. This identifier is advertised to connected devices as part of the Link Aggregation Control Protocol (LACP) to create a unified view of the MLAG pair.

```
Leaf-2a(config)#show running-configuration interfaces vxlan 1
interface Vxlan1
   vxlan source-interface Loopback0
   vxlan virtual-router encapsulation mac-address mlag-system-id
   vxlan udp-port 4789
   vxlan vlan 11 vni 10011
   vxlan vlan 12 vni 10012
   vxlan vlan 13 vni 10013
   vxlan mlag source-interface Loopback1
```

An example of the host facing portchannel interface on Leaf2a:

```
Leaf-2a(config)#show running-configuration interfaces po2
interface Port-Channel2
   switchport trunk allowed vlan 11-13
   switchport mode trunk
   mlag 2
Leaf-2a(config)#show run interfaces et6
```

```
interface Ethernet6
   channel-group 2 mode active
```

Underlay and Overlay Configuration and Validation of Junos OS and Arista's EOS Switches

Below is an example for the EBGP overlay control plane configuration from Leaf2a to Spine1 and Spine2, the same is used for all Arista's EOS leaf switches.

```
Leaf-2a(config)#show running-configuration section bgp
*snip*
router bgp 65002
   router-id 151.0.0.2
   no bgp default ipv4-unicast
   distance bgp 20 100 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor 151.0.0.11 peer group EVPN-OVERLAY
   neighbor 151.0.0.11 remote-as 65011
   neighbor 151.0.0.11 description Spine-1
   neighbor 151.0.0.12 peer group EVPN-OVERLAY
   neighbor 151.0.0.12 remote-as 65011
   neighbor 151.0.0.12 description Spine-2
   address-family evpn
      neighbor EVPN-OVERLAY activate
```

The BGP EVPN Underlay config is the same as shown for the spines earlier.

Below is an output from Leaf2a summarizing the EBGP overlay peering towards Spines using loopback0

```
Leaf-2a(config)#show bgp evpn summary
BGP summary information for VRF default
Router identifier 151.0.0.2, local AS number 65002
Neighbor Status Codes: m - Under maintenance
  Description            Neighbor   V AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State   PfxRcd PfxAcc
  Spine-1                151.0.0.11 4 65011           2076      1970    0    0    1d00h Estab   55     55
  Spine-2                151.0.0.12 4 65011           2056      2024    0    0    1d01h Estab   55     55
```

On Junos OS devices, loopback0 is configured and used for EBGP overlay peering. Below is an output from Leaf7, which can be used for all Junos OS switches.

```
root@Leaf-7# show interfaces lo0
unit 0 {
    family inet {
        address 3.1.1.7/32;
    }
}
```

For Junos OS, 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.

```
root@Leaf-7# show protocols bgp
group UNDERLAY {
    type external;
    export UNDERLAY_EXPORT;
    local-as 65007;
    multipath {
        multiple-as;
    }
    neighbor 161.0.1.18 {
        description SPINE-1;
        peer-as 65011;
    }
    neighbor 161.0.1.26 {
        description SPINE-2;
        peer-as 65011;
    }
}
group OVERLAY {
    type external;
    multihop;
    local-address 3.1.1.7;
    family evpn {
        signaling;
    }
    local-as 65007;
    multipath;
    neighbor 151.0.0.11 {
        description SPINE-1;
        local-address 3.1.1.7;
        peer-as 65011;
    }
    neighbor 151.0.0.12 {
        description SPINE-2;
        local-address 3.1.1.7;
        peer-as 65011;
    }
    vpn-apply-export;
}
```

Below is the output validating the underlay and overlay peering of Leaf7, which is a Junos OS device.

```
root@Leaf-7> show bgp summary

Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.evpn.0
                     108         54          0          0          0          0
inet.0
                      24         24          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
151.0.0.11            65011       3108       2766       0       3    20:55:03 Establ
  bgp.evpn.0: 52/54/54/0
  __default_evpn__.evpn.0: 0/0/0/0
  V11_MACVRF.evpn.0: 23/23/23/0
  VRF1_MAC_VRF_AWARE.evpn.0: 29/31/31/0
151.0.0.12            65011       3112       2765       0       2    20:55:01 Establ
  bgp.evpn.0: 2/54/54/0
  __default_evpn__.evpn.0: 0/0/0/0
  V11_MACVRF.evpn.0: 0/23/23/0
```

```
  VRF1_MAC_VRF_AWARE.evpn.0: 2/31/31/0
161.0.1.18            65011       2971       2762       0       3    20:55:04 Establ
  inet.0: 12/12/12/0
161.0.1.26            65011       2966       2762       0       2    20:55:04 Establ
  inet.0: 12/12/12/0
```

MAC-VRF/L2VPN Configuration and Validation on Arista's EOS and Junos OS Devices

First, VLANs need to be created for L2 forwarding. Vlans 11,12 and 13 is configured for learning purposes but the end-to-end connectivity is demonstrated only for vlan11 in this section. Below is the config of standalone Arista's EOS device - Leaf-1.

```
Leaf-1(config)#show running-configuration section vlan
*snip*
vlan 11
   name VLAN11
!
vlan 12
   name VLAN12
!
vlan 13
   name VLAN13
```

UDP port 4789 is the default value for the VXLAN packet. For Leaf1 interface loopback1 is used as a source interface for the inner VXLAN packet and as the next-hop address in the NLRI for any BGP updates. VXLAN to VNI mapping is configured under interface vxlan1 as shown below.

```
Leaf-1(config)#show running-configuration section vxlan1
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 11 vni 10011
   vxlan vlan 12 vni 10012
   vxlan vlan 13 vni 10013
```

For the sake of consistency across all VTEPs that provide L2VPN connectivity, the VLAN to VNI mapping should be same.

MAC-VRF instance is a Layer2 virtual routing and forwarding instance used to segment and isolate Ethernet MAC address spaces in the network. It provides tenant or service isolation and isolates BUM traffic to its specific Layer 2 domain by restricting it from being forwarded to another MAC-VRF. This in return reduces congestion and improves performance.
BUM traffic is encapsulated in VXLAN packets and forwarded to only the relevant VTEPs that are part of the same MAC-VRF.

Below is an example of a VLAN based and VLAN aware MAC-VRF on Leaf1.
The VLAN based config starts with the name of the vlan (vlan 11) and followed by the RD and RT which is used for import and export.
The "redistribute learned" command enables the VTEP to advertise dynamically learned MAC addresses in VLAN 11 into the EVPN control plane.
The VLAN aware config also lists the RD, RT and the list of VLANs under the MAC-VRF config.

```
Leaf-1(config)#show running-configuration section bgp
*snip*
   vlan 11
      rd 3.1.1.1:10011
      route-target both 10011:10011
      redistribute learned
   !
   vlan-aware-bundle VRF1_MAC_VRF_AWARE
      rd 3.1.1.1:1213
      route-target both 1213:1213
      redistribute learned
      vlan 12-13
```

Even in a MLAG pair the RD should be unique as it simplifies Convergence and Redundancy, for instance when one MLAG peer goes offline, the other peer continues to advertise the same MAC routes with its own RD. It also helps with enabling ECMP as the load is distributed between the two MLAG peers as the spines will have two different routes for the same mac-address.

Below is an example of MAC-VRF config on MLAG pair Leaf2a and Leaf2b

```
Leaf-2a(config)#show running-configuration section
bgp
router bgp 65002
vlan 11
      rd 3.1.1.2:10011
      route-target both 10011:10011
      redistribute learned
   !
   vlan-aware-bundle VRF1_MAC_VRF_AWARE
      rd 3.1.1.2:1213
      route-target both 1213:1213
      redistribute learned
      vlan 12-13
```

```
Leaf-2b(config)#show running-configuration section
bgp
router bgp 65002
   vlan 11
      rd 3.1.2.2:10011
      route-target both 10011:10011
      redistribute learned
   !
   vlan-aware-bundle VRF1_MAC_VRF_AWARE
      rd 3.1.2.2:1213
      route-target both 1213:1213
      redistribute learned
      vlan 12-13
```

Validating L2VPN Service with Type-2 and Type-3 Routes

Let's validate L2VPN Service with Type-2 and Type-3 routes on Leaf2a Leaf2b and Leaf7 for vlan-id 11.
Type-3 imet route on Arista's EOS and Junos OS.

Below is the type-3 route on Leaf2a received from Leaf7. It lists the VNI ids, PMSI tunnel which is "Ingress Replication" and so on.

```
Leaf-2a#show bgp evpn route-type imet rd 3.1.1.7:10011 detail
BGP routing table information for VRF default
Router identifier 151.0.0.2, local AS number 65002
BGP routing table entry for imet 3.1.1.7, Route Distinguisher: 3.1.1.7:10011
 Paths: 2 available
  65011 65007
    3.1.1.7 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011
      PMSI Tunnel: Ingress Replication, MPLS Label: 10011, Leaf Information Required: false, Tunnel ID: 3.1.1.7
  65011 65007
    3.1.1.7 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011
      PMSI Tunnel: Ingress Replication, MPLS Label: 10011, Leaf Information Required: false, Tunnel ID: 3.1.1.7
```

This is the Junos OS equivalent of Type3 route on Leaf7 received from Leaf2a and 2b (MLAG pair).

```
root@Leaf-7> show route table V11_MACVRF.evpn.0 match-prefix *3:3.1.1.2*

V11_MACVRF.evpn.0: 26 destinations, 49 routes (26 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3:3.1.1.2:10011::0::3.1.1.2/248 IM
                   *[BGP/170] 1d 19:43:47, localpref 100, from 151.0.0.11
                      AS path: 65011 65002 I, validation-state: unverified
                       to 161.0.1.18 via ge-0/0/0.0
                    >  to 161.0.1.26 via ge-0/0/1.0
                    [BGP/170] 1d 19:43:45, localpref 100, from 151.0.0.12
                      AS path: 65011 65002 I, validation-state: unverified
                       to 161.0.1.18 via ge-0/0/0.0
                    >  to 161.0.1.26 via ge-0/0/1.0
```

Type2 MAC IP routes on Arista's EOS and Junos OS.

Let's validate the local and remotely learned MAC address on vlan 11.

```
Leaf-2a#show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et7, Et8, PEt5, PEt7, PEt8, Po1
11    VLAN11                           active    Po1, Po2, Vx1
12    VLAN12                           active    Po1, Po2, Vx1
13    VLAN13                           active    Po1, Po2, Vx1
4093  VLAN4093                         active    Cpu, Po1
4094  VLAN4094                         active    Cpu, Po1
```

The logical VTEP interface is vxlan1 which lists lo0 as the source interface for single-homed hosts and lo1 for dual homed as explained earlier. HeadEnd Replication (HER) is used for handling BUM Traffic in Arista's EOS devices.
Static VLAN to VNI mapping is listed as per the configuration.
Vlan11, 12 and 13 HER flood list is also mentioned in this output.

```
Leaf-2a#show interfaces vxlan1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback0 and is active with 151.0.0.2
  MLAG Source interface is Loopback1 and is active with 3.1.1.2
  Listening on UDP port 4789
  Replication/Flood Mode is headend with Flood List Source: EVPN
  Remote MAC learning via EVPN
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
    [11, 10011]        [12, 10012]        [13, 10013]
  Note: All Dynamic VLANs used by VCS are internal VLANs.
        Use 'show vxlan vni' for details.
  Static VRF to VNI mapping is not configured
  Headend replication flood vtep list is:
    11 3.1.1.6         3.1.1.3         3.1.1.7         3.1.1.1         3.1.1.4
       3.1.1.5
    12 3.1.1.6         3.1.1.3         3.1.1.7         3.1.1.1         3.1.1.4
       3.1.1.5
    13 3.1.1.6         3.1.1.3         3.1.1.7         3.1.1.1         3.1.1.4
       3.1.1.5
  MLAG Shared Router MAC is 5200.0045.abdf
```

The flood list for vlan 11 includes both Arista's EOS (Leaf1, 3, 4) and JUNOS (Leaf5 ,6 ,7) devices.

```
Leaf-2a#show vxlan flood vtep vlan 11
          VXLAN Flood VTEP Table
--------------------------------------------------------------------------------

VLANS                            Ip Address
-----------------------------   ------------------------------------------------
11                              3.1.1.1         3.1.1.3         3.1.1.4
                                3.1.1.5         3.1.1.6         3.1.1.7
```

Let's pick one mac-address 2c6b.f5bb.f0f0, this belongs to Host5 connected to Leaf5 and Leaf6 (Junos OS). Below output reflects the Type2 Mac+IP route received from 3.1.1.5 and 3.1.1.6.

```
Leaf-2a#show bgp evpn route-type mac-ip 2c6b.f5bb.f0f0
BGP routing table information for VRF default
Router identifier 151.0.0.2, local AS number 65002
Route status codes: * - valid, > - active, S - Stale, E - ECMP head, e - ECMP
                    c - Contributing to ECMP, % - Pending BGP convergence
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

          Network                Next Hop              Metric  LocPref Weight  Path
 * >Ec    RD: 3.1.1.5:10011 mac-ip 2c6b.f5bb.f0f0
                                 3.1.1.5               -       100     0       65011 65005 i
 *  ec    RD: 3.1.1.5:10011 mac-ip 2c6b.f5bb.f0f0
                                 3.1.1.5               -       100     0       65011 65005 i
 * >Ec    RD: 3.1.1.6:10011 mac-ip 2c6b.f5bb.f0f0
                                 3.1.1.6               -       100     0       65011 65006 i
 *  ec    RD: 3.1.1.6:10011 mac-ip 2c6b.f5bb.f0f0
                                 3.1.1.6               -       100     0       65011 65006 i
 * >Ec    RD: 3.1.1.5:10011 mac-ip 2c6b.f5bb.f0f0 11.0.0.5
                                 3.1.1.5               -       100     0       65011 65005 i
 *  ec    RD: 3.1.1.5:10011 mac-ip 2c6b.f5bb.f0f0 11.0.0.5
                                 3.1.1.5               -       100     0       65011 65005 i
 * >Ec    RD: 3.1.1.6:10011 mac-ip 2c6b.f5bb.f0f0 11.0.0.5
                                 3.1.1.6               -       100     0       65011 65006 i
 *  ec    RD: 3.1.1.6:10011 mac-ip 2c6b.f5bb.f0f0 11.0.0.5
                                 3.1.1.6               -       100     0       65011 65006 i
```

Here is the detailed output of Type-2 route which includes:

- Next-Hop as 3.1.1.5 and 3.1.1.6
- Route-Target as Route-Target-AS:10011:10011
- VNI as 10011
- ESI as 0000:0000:0000:0000:5555
- Tunnel Type as TunnelEncap:tunnelTypeVxlan

```
Leaf-2a#show bgp evpn route-type mac-ip 2c6b.f5bb.f0f0 detail
BGP routing table information for VRF default
Router identifier 151.0.0.2, local AS number 65002
BGP routing table entry for mac-ip 2c6b.f5bb.f0f0, Route Distinguisher: 3.1.1.5:10011
 Paths: 2 available
  65011 65005
    3.1.1.5 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
  65011 65005
    3.1.1.5 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
BGP routing table entry for mac-ip 2c6b.f5bb.f0f0, Route Distinguisher: 3.1.1.6:10011
 Paths: 2 available
  65011 65006
    3.1.1.6 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
  65011 65006
    3.1.1.6 from 151.0.0.12 (151.0.0.12)
```

```
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
BGP routing table entry for mac-ip 2c6b.f5bb.f0f0 11.0.0.5, Route Distinguisher: 3.1.1.5:10011
 Paths: 2 available
  65011 65005
    3.1.1.5 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan EvpnNdFlags:pflag
      Rx path id: 0x1
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
  65011 65005
    3.1.1.5 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan EvpnNdFlags:pflag
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
BGP routing table entry for mac-ip 2c6b.f5bb.f0f0 11.0.0.5, Route Distinguisher: 3.1.1.6:10011
 Paths: 2 available
  65011 65006
    3.1.1.6 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
  65011 65006
    3.1.1.6 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
```

This output lists the remotely learned mac address which are part of Vlan11. This will populate as and when Type2 routes are received/accepted from the remote VTEPs.
For example, MAC address 2c6b.f5bb.f0f0 is learnt from remote VTEP 3.1.1.5 and 3.1.1.6 which is loopback0 IP address for Leaf5 and Leaf6 which are part of an ESI-LAG. Note both the devices are running Junos OS.

```
Leaf-2a#show vxlan address-table vlan 11
          Vxlan Mac Address Table
----------------------------------------------------------------------

VLAN  Mac Address     Type      Prt  VTEP             Moves   Last Move
----  -----------     ----      ---  ----             -----   ---------
  11  2c6b.f5bb.f0f0  EVPN      Vx1  3.1.1.5          1       1 day, 4:32:36 ago
                                     3.1.1.6
  11  2c6b.f5c1.2df0  EVPN      Vx1  3.1.1.7          1       1 day, 1:26:31 ago
  11  5000.006b.2e70  EVPN      Vx1  3.1.1.1          1       1 day, 1:19:39 ago
  11  5000.00af.d3f6  EVPN      Vx1  3.1.1.3          1       1 day, 4:32:34 ago
                                     3.1.1.4
Total Remote Mac Addresses for this criterion: 4
```

Local MAC address table for Vlan11 on Leaf2a which includes remote MAC addresses as well.

```
Leaf-2a#show mac address-table vlan 11
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
  11    2c6b.f5bb.f0f0    DYNAMIC     Vx1        1       1 day, 4:33:22 ago
  11    2c6b.f5c1.2df0    DYNAMIC     Vx1        1       1 day, 1:27:17 ago
  11    5000.006b.2e70    DYNAMIC     Vx1        1       1 day, 1:20:25 ago
  11    5000.00af.d3f6    DYNAMIC     Vx1        1       1 day, 4:33:20 ago
  11    5000.00ba.c6f8    DYNAMIC     Po2        1       1 day, 4:33:12 ago
```

```
Total Mac Addresses for this criterion: 5
```

Similarly, let's validate Type2 routes on Junos OS devices for vlan11 on Leaf7. Let's track the control plane flow for mac address 5000.00ba.c6f8 (Host3) which is locally learnt on Leaf2a (Arista's EOS).

```
root@Leaf-7> show vlans
Routing instance        VLAN name             Tag          Interfaces
V11_MACVRF              VLAN11                11
                                                           esi.712*
                                                           esi.713*
                                                           ge-0/0/2.11*
                                                           vtep.32770*
                                                           vtep.32772*
                                                           vtep.32773*
                                                           vtep.32776*
                                                           vtep.32779*
                                                           vtep.32780*
```

This output lists the source and all the available remote VTEP's in this fabric.

```
root@Leaf-7> show interfaces vtep | grep "VXLAN Endpoint Address:"
    VXLAN Endpoint Type: Source, VXLAN Endpoint Address: 3.1.1.7, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Source, VXLAN Endpoint Address: 3.1.1.7, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.6, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.6, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.5, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.3, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.5, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.3, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.4, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.4, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.2, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.2, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.1, L2 Routing Instance: V11_MACVRF, L3 Routing
Instance: default
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 3.1.1.1, L2 Routing Instance: VRF1_MAC_VRF_AWARE, L3
Routing Instance: default
```

There are different processes involved in Junos OS. The result is that the bgp.evpn.0 table on Leaf7 should have Host3's MAC address.

```
root@Leaf-7> show route table bgp.evpn.0 evpn-mac-address 5000.00ba.c6f8
bgp.evpn.0: 63 destinations, 117 routes (63 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:3.1.1.2:10011::0::50:00:00:ba:c6:f8/304 MAC/IP
                   *[BGP/170] 1d 18:18:38, localpref 100, from 151.0.0.11
                      AS path: 65011 65002 I, validation-state: unverified
                       to 161.0.1.18 via ge-0/0/0.0, Push 10011
                    >  to 161.0.1.26 via ge-0/0/1.0, Push 10011
                    [BGP/170] 1d 18:18:36, localpref 100, from 151.0.0.12
                      AS path: 65011 65002 I, validation-state: unverified
                       to 161.0.1.18 via ge-0/0/0.0, Push 10011
                    >  to 161.0.1.26 via ge-0/0/1.0, Push 10011
```

From the bgp.evpn.table, this gets pushed into the table specific to the routing-instance and eventually the ethernet-switching table. The protocol next-hop for the Host3's mac address is 3.1.1.2 which is the VTEP IP for Leaf2a and 2b (MLAG).

```
root@Leaf-7> show route table V11_MACVRF.evpn.0 evpn-mac-address 5000.00ba.c6f8 extensive

V11_MACVRF.evpn.0: 26 destinations, 49 routes (26 active, 0 holddown, 0 hidden)
2:3.1.1.2:10011::0::50:00:00:ba:c6:f8/304 MAC/IP (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 3.1.1.2:10011
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7ca5c94
                Next-hop reference count: 8
                Kernel Table Id: 0
                Source: 151.0.0.11
                Protocol next hop: 3.1.1.2
                Label operation: Push 10011
                Label TTL action: prop-ttl
                Load balance label: Label 10011: None;
                Indirect next hop: 0x2 no-forward INH Session ID: 0
                Indirect next hop: INH non-key opaque: 0x0 INH key opaque: 0x0
                State: <Secondary Active Ext>
                Local AS: 65007 Peer AS: 65011
                Age: 1d 18:23:12        Metric2: 0
                Validation State: unverified
                Task: BGP_65011.151.0.0.11
                Announcement bits (1): 0-V11_MACVRF-evpn
                AS path: 65011 65002 I
                Communities: target:10011:10011 encapsulation:vxlan(0x8)
                Import Accepted
                Route Label: 10011
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 151.0.0.11
                Primary Routing Table: bgp.evpn.0
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 3.1.1.2 ResolvState: Resolved
                        Label operation: Push 10011
                        Label TTL action: prop-ttl
                        Load balance label: Label 10011: None;
                        Indirect next hop: 0x2 no-forward INH Session ID: 0
                        Indirect next hop: INH non-key opaque: 0x0 INH key opaque: 0x0
                        Indirect path forwarding next hops: 2
                                Next hop type: Router
                                Next hop: 161.0.1.18 via ge-0/0/0.0
                                Session Id: 0
                                Next hop: 161.0.1.26 via ge-0/0/1.0
                                Session Id: 0
                                3.1.1.2/32 Originating RIB: inet.0
                                  Node path count: 1
                                  Forwarding nexthops: 2
                                        Next hop type: Router
                                        Next hop: 161.0.1.18 via ge-0/0/0.0
                                        Session Id: 0
                                        Next hop: 161.0.1.26 via ge-0/0/1.0
                                        Session Id: 0
         *snip*
```

```
root@Leaf-7> show evpn database mac-address 5000.00ba.c6f8
Instance: V11_MACVRF
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     10011      50:00:00:ba:c6:f8  3.1.1.2                        Jan 07 00:37:59
```

```
root@Leaf-7> show ethernet-switching table vlan-id 11 50:00:00:ba:c6:f8

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


Ethernet switching table : 5 entries, 5 learned
Routing instance : V11_MACVRF
   Vlan                MAC                 MAC       GBP    Logical                 SVLBNH/      Active
   name                address             flags     tag    interface               VENH Index   source
   VLAN11              50:00:00:ba:c6:f8   DR               vtep.32779                           3.1.1.2
```

At the end of this, host-3 (Arista's EOS) should now be able to ping host-6 (Junos OS):

```
Host3#ping 11.0.0.6
PING 11.0.0.6 (11.0.0.6) 72(100) bytes of data.
80 bytes from 11.0.0.6: icmp_seq=1 ttl=64 time=24.3 ms
80 bytes from 11.0.0.6: icmp_seq=2 ttl=64 time=23.5 ms
80 bytes from 11.0.0.6: icmp_seq=3 ttl=64 time=20.7 ms
80 bytes from 11.0.0.6: icmp_seq=4 ttl=64 time=24.4 ms
80 bytes from 11.0.0.6: icmp_seq=5 ttl=64 time=24.2 ms

--- 11.0.0.6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 70ms
rtt min/avg/max/mdev = 20.746/23.472/24.414/1.405 ms, pipe 3, ipg/ewma 17.742/23.944 ms
```

Bridged Overlay with Arista's EOS ESI-LAG and Junos OS ESI-LAG

Leaf3a and 3b (Arista's EOS) are in an ESI-LAG along with Leaf 5a and 5b (Junos OS).

To understand how ESI is implemented on Arista's EOS devices lets dive into the config part of Leaf3a and 3b. As we can see, Host4 is part of portchannel3 (ESI) which has the same ESI ID (ethernet segment identifier) and LACP system-id on both leafs.
Route-target import needs to be configured on both ESI pairs for importing type-4 routes between the peers.

```
Leaf-3a(config)#show run interfaces port-Channel 3
interface Port-Channel3
   switchport trunk allowed vlan 11-13
   switchport mode trunk
   !
   evpn ethernet-segment
      identifier 0000:0000:0000:0333:3333
      route-target import 00:00:03:33:33:33
   lacp system-id 0000.0333.3333
```

```
Leaf-3b(config)#show run interfaces port-Channel 3
interface Port-Channel3
   switchport trunk allowed vlan 11-13
   switchport mode trunk
   !
   evpn ethernet-segment
      identifier 0000:0000:0000:0333:3333
      route-target import 00:00:03:33:33:33
   lacp system-id 0000.0333.3333
```

In Junos route-target import is not needed as we use a default RT of 00:00:00:00:00:00 for importing Type4 routes.
A default policy is automatically created as soon as an ESI is configured where the es-import-target is 0 [es-import-target:0-0-0-0-0-0].

```
root@Leaf-5a# show interfaces ae2
flexible-vlan-tagging;
mtu 9100;
encapsulation extended-vlan-bridge;
esi {
    00:00:00:00:00:00:00:00:55:55;
    all-active;
}
aggregated-ether-options {
    lacp {
        active;
        system-id 00:00:00:55:55:55;
    }
```

```
root@Leaf-5b# show interfaces ae2
flexible-vlan-tagging;
mtu 9100;
encapsulation extended-vlan-bridge;
esi {
    00:00:00:00:00:00:00:00:55:55;
    all-active;
}
aggregated-ether-options {
    lacp {
        active;
        system-id 00:00:00:55:55:55;
    }
```

```
}
unit 11 {
    vlan-id 11;
}
unit 12 {
    vlan-id 12;
}
unit 13 {
    vlan-id 13;
}
```

```
}
unit 11 {
    vlan-id 11;
}
unit 12 {
    vlan-id 12;
}
unit 13 {
    vlan-id 13;
}
```

```
root@Leaf-5a> show policy __vrf-import-__default_evpn__-internal__
Policy __vrf-import-__default_evpn__-internal__:  [EVPN_ESI/]
    Term unnamed:
        from community __vrf-community-__default_evpn__-import-internal__ [es-import-target:0-0-0-0-0-0 ]
        then accept
    Term unnamed:
        then reject
```

Validating L2VPN Service with Type1 and Type4 Routes

Below is the output from Leaf3a where it lists Type1 routes that is received per VNI, in our case which is 10011 for vlan11 from Leaf5a (Junos OS).

```
Leaf-3a#show bgp evpn route-type auto-discovery rd 3.1.1.5:10011 detail
BGP routing table information for VRF default
Router identifier 151.0.0.4, local AS number 65003
BGP routing table entry for auto-discovery 0 0000:0000:0000:0000:5555, Route Distinguisher: 3.1.1.5:10011
 Paths: 2 available
  65011 65005
    3.1.1.5 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011
  65011 65005
    3.1.1.5 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011
```

This is the Type4 routes received on Leaf3a from Leaf5a. It lists the EvpnEsImportRt as 0 which was explained earlier.

```
Leaf-3a#show bgp evpn route-type ethernet-segment rd 3.1.1.5:0 detail
BGP routing table information for VRF default
Router identifier 151.0.0.4, local AS number 65003
BGP routing table entry for ethernet-segment 0000:0000:0000:0000:5555 3.1.1.5, Route Distinguisher: 3.1.1.5:0
 Paths: 2 available
  65011 65005
    3.1.1.5 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: TunnelEncap:tunnelTypeVxlan EvpnEsImportRt:00:00:00:00:00:00
  65011 65005
    3.1.1.5 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: TunnelEncap:tunnelTypeVxlan EvpnEsImportRt:00:00:00:00:00:00
      Rx path id: 0x1
```

Below output shows the EVPN instance for vlan11 listing the elected Designated forwarder (DF), which is Leaf3b (3.1.1.4) in this ESI pair.

```
Leaf-3a#show bgp evpn instance vlan 11
EVPN instance: VLAN 11
  Route distinguisher: 3.1.1.3:10011
  Route target import: Route-Target-AS:10011:10011
  Route target export: Route-Target-AS:10011:10011
  Service interface: VLAN-based
  Local VXLAN IP address: 3.1.1.3
  VXLAN: enabled
  MPLS: disabled
  Remote ethernet segment:
    ESI: 0000:0000:0000:0000:5555
    Active TEPs: 3.1.1.5, 3.1.1.6
  Local ethernet segment:
    ESI: 0000:0000:0000:0333:3333
      Interface: Port-Channel3
      Mode: all-active
      State: up
      ES-Import RT: 00:00:03:33:33:33
      DF election algorithm: modulus
      Designated forwarder: 3.1.1.4
      Non-Designated forwarder: 3.1.1.3
```

This mac address 2c6b.f5bb.f0f0 is locally learnt on Leaf5a, below is the Type2 route received on Leaf3a with ESI 0000:0000:0000:0000:5555 for VLAN11.

```
Leaf-3a#show bgp evpn route-type mac-ip 2c6b.f5bb.f0f0 vni 10011 detail
BGP routing table information for VRF default
Router identifier 151.0.0.4, local AS number 65003
BGP routing table entry for mac-ip 2c6b.f5bb.f0f0, Route Distinguisher: 3.1.1.5:10011
 Paths: 2 available
  65011 65005
    3.1.1.5 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
  65011 65005
    3.1.1.5 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
BGP routing table entry for mac-ip 2c6b.f5bb.f0f0, Route Distinguisher: 3.1.1.6:10011
 Paths: 2 available
  65011 65006
    3.1.1.6 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
  65011 65006
    3.1.1.6 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:10011:10011 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:5555
```

VXLAN address table on Leaf3a showing the VTEP endpoints for this mac address 3.1.1.5 and 3.1.1.6 which belongs to Leaf5a and 5b.

```
Leaf-3a#show vxlan address-table address 2c6b.f5bb.f0f0 vlan 11
          Vxlan Mac Address Table
----------------------------------------------------------------------
```

```
VLAN  Mac Address     Type      Prt  VTEP             Moves   Last Move
----  -----------     ----      ---  ----             -----   ---------
  11  2c6b.f5bb.f0f0  EVPN      Vx1  3.1.1.5          1       2 days, 0:19:02 ago
                                     3.1.1.6
Total Remote Mac Addresses for this criterion: 1
```

This MAC address is now installed in the Local MAC address table of Leaf3a.

```
Leaf-3a#show mac address-table address 2c6b.f5bb.f0f0 vlan 11
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
  11    2c6b.f5bb.f0f0    DYNAMIC     Vx1        1       2 days, 0:19:11 ago
Total Mac Addresses for this criterion: 1
```

In Junos OS Leaf5a Type1 routes received from ESI pair Leaf3a and 3b.

```
root@Leaf-5a> show route table V11_MACVRF.evpn.0 match-prefix *1:*

V11_MACVRF.evpn.0: 25 destinations, 46 routes (25 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:3.1.1.3:1::03333333::FFFF:FFFF/192 AD/ESI
                   *[BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.11
                      AS path: 65011 65003 I, validation-state: unverified
                       to 161.0.1.4 via ge-0/0/0.0
                    >  to 161.0.1.14 via ge-0/0/1.0
                    [BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.12
                      AS path: 65011 65003 I, validation-state: unverified
                       to 161.0.1.4 via ge-0/0/0.0
                    >  to 161.0.1.14 via ge-0/0/1.0
1:3.1.1.3:10011::03333333::0/192 AD/EVI
                   *[BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.11
                      AS path: 65011 65003 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0, Push 10011
                       to 161.0.1.14 via ge-0/0/1.0, Push 10011
                    [BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.12
                      AS path: 65011 65003 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0, Push 10011
                       to 161.0.1.14 via ge-0/0/1.0, Push 10011
1:3.1.1.4:1::03333333::FFFF:FFFF/192 AD/ESI
                   *[BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.11
                      AS path: 65011 65004 I, validation-state: unverified
                       to 161.0.1.4 via ge-0/0/0.0
                    >  to 161.0.1.14 via ge-0/0/1.0
                    [BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.12
                      AS path: 65011 65004 I, validation-state: unverified
                       to 161.0.1.4 via ge-0/0/0.0
                    >  to 161.0.1.14 via ge-0/0/1.0
1:3.1.1.4:10011::03333333::0/192 AD/EVI
                   *[BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.11
                      AS path: 65011 65004 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0, Push 10011
                       to 161.0.1.14 via ge-0/0/1.0, Push 10011
                    [BGP/170] 2d 00:01:22, localpref 100, from 151.0.0.12
                      AS path: 65011 65004 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0, Push 10011
                       to 161.0.1.14 via ge-0/0/1.0, Push 10011
```

Type4 routes on Leaf5a from Leaf3a and 3b

```
root@Leaf-5a> show route table bgp.evpn.0 match-prefix *4:*
4:3.1.1.3:1::03333333:3.1.1.3/296 ES
                   *[BGP/170] 00:19:33, localpref 100, from 151.0.0.12
                      AS path: 65011 65003 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0
                       to 161.0.1.14 via ge-0/0/1.0
                    [BGP/170] 00:19:33, localpref 100, from 151.0.0.11
                      AS path: 65011 65003 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0
                       to 161.0.1.14 via ge-0/0/1.0
4:3.1.1.4:1::03333333:3.1.1.4/296 ES
                   *[BGP/170] 00:30:33, localpref 100, from 151.0.0.11
                      AS path: 65011 65004 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0
                       to 161.0.1.14 via ge-0/0/1.0
                    [BGP/170] 00:30:33, localpref 100, from 151.0.0.12
                      AS path: 65011 65004 I, validation-state: unverified
                    >  to 161.0.1.4 via ge-0/0/0.0
                       to 161.0.1.14 via ge-0/0/1.0
```

Asymmetric IRB on Arista's EOS and Asymmetric IRB on Junos OS

Overview and Topology

For the Asymmetric IRB on Arista's EOS and Junos OS use case, we will use the following topology:

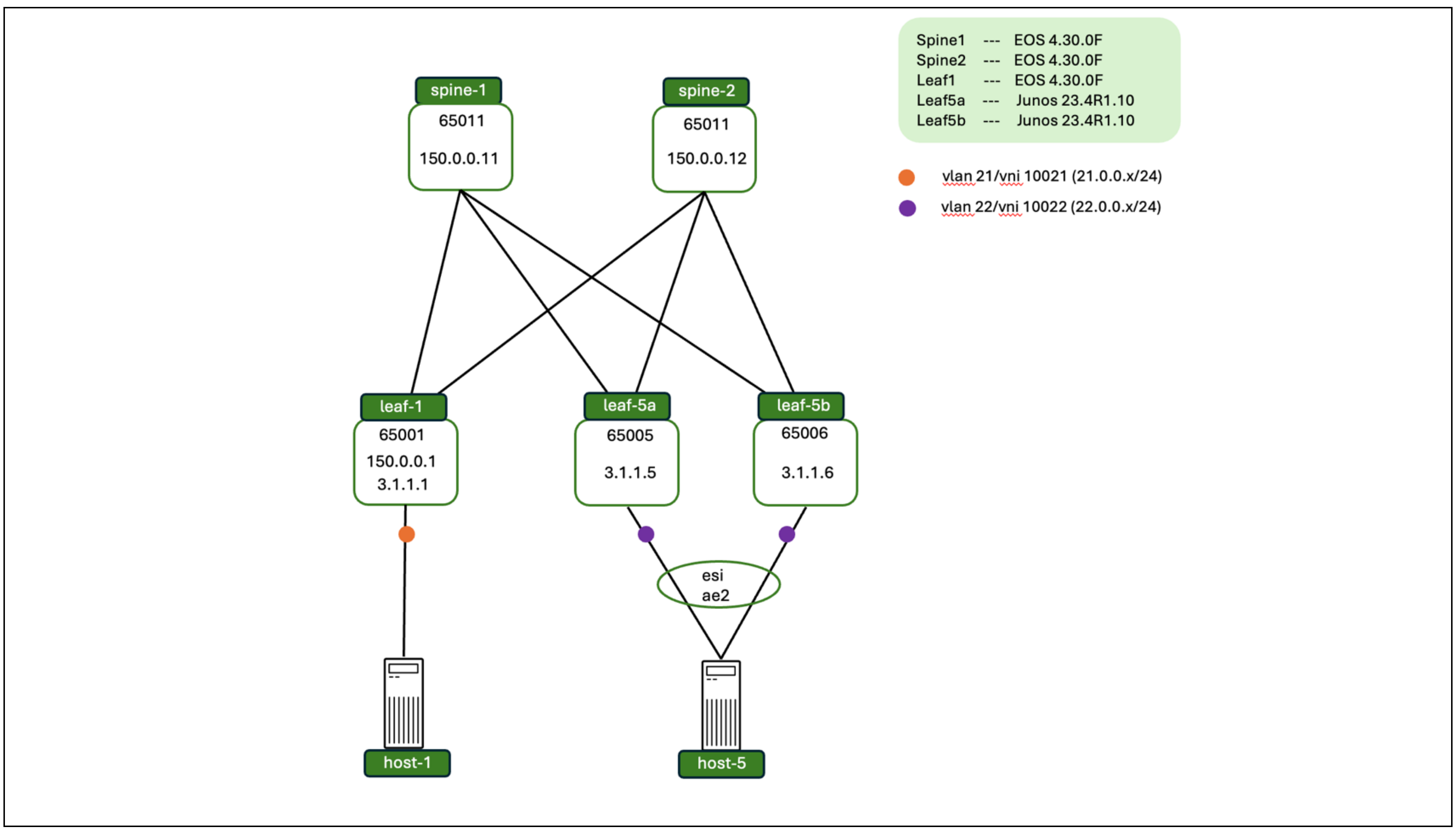

Figure 2 : Topology Diagram for Asymmetric IRB on Arista's EOS and Junos OS

In Arista's EOS asymmetric IRB model, the ingress VTEP performs inter-subnet routing and VXLAN-bridges the packet to the destination VTEP. The egress VTEP removes the VXLAN header and forwards the packet to the local Layer 2 domain based on VNI-to-VLAN mapping. On the return path, the destination VTEP takes over routing and bridging, making the process asymmetric.
Each VTEP learns host MAC addresses and MAC-to-IP bindings through local ARP snooping and type-2 route advertisements from other VTEPs. Inter-subnet routing for all subnets on all VTEPs is achieved using an anycast IP address configured per subnet on each VTEP. This anycast IP serves as the hosts' default gateway, enabling any connected VTEP to act as the gateway regardless of the host's location.

When traffic flows from Host-1 in VLAN-21 to Host-5 in VLAN-22, the ingress VTEP (Leaf1) routes the packet locally into subnet-22/VNI 10022. It then encapsulates the packet using VXLAN, adding VNI 10022 to the VXLAN header and setting Host-5's MAC address as the inner destination MAC. At the receiving VTEP (Leaf5a/5b), only a local Layer 2 lookup is required, using the VNI-to-VLAN mapping to resolve Host-5's MAC address.

Data Plane Flow in Asymmetric IRB

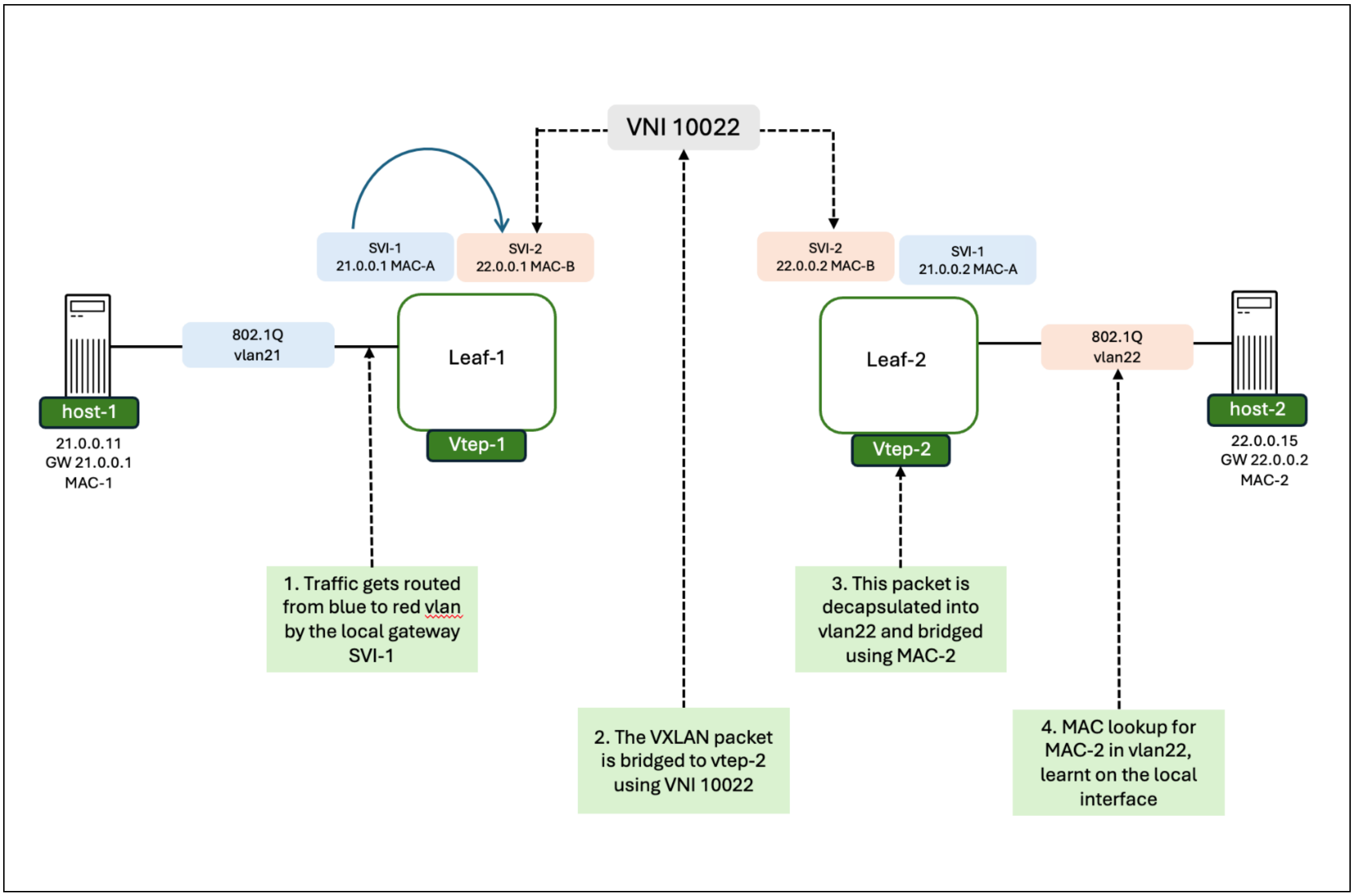

Figure 3 : Asymmetric Data Plane Flow Diagram

Configuring Asymmetric IRB on Arista's EOS

Leaf1 will have to be configured with IRB interfaces for VLAN21 and 22 along with their respective VNI's.

```
Leaf-1#show running-configuration section interface
*snip*
interface Ethernet3
   switchport trunk allowed vlan 21
   switchport mode trunk
!
```

```
interface Vlan21
   mtu 9200
   ip address 21.0.0.1/24
!
interface Vlan22
   mtu 9200
   ip address 22.0.0.1/24
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 21 vni 10021
   vxlan vlan 22 vni 10022
```

In this example, both the VLANs are part of a MAC-VRF VLAN-AWARE bundle.

```
Leaf-1#show running-configuration section bgp
*snip*
router bgp 65001
   vlan-aware-bundle VRF2_MAC_VRF_AWARE
      rd 3.1.1.1:2122
      route-target both 2122:2122
      redistribute learned
      vlan 21-22
*snip*
```

Configuring Asymmetric IRB on Junos OS

In Junos OS, we configure leaf5a and 5b for Asymmetric IRB. For this, we require that the IRB interface for both VLAN21 and VLAN22 exist on both leafs, along with their corresponding L2VNIs. Routing-instances are created for both VLAN21 and VLAN21 with service-type VLAN-aware.

```
root@Leaf-5a# show interfaces
*snip*

irb {
    unit 21 {
        family inet {
            address 21.0.0.1/24;
        }
    }
    unit 22 {
        family inet {
            address 22.0.0.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 3.1.1.5/32;
        }
    }
policy-options {
    prefix-list adv_loopback {
        3.1.1.0/24;
    }
    policy-statement UNDERLAY_EXPORT {
        term 1 {
            from {
                route-filter 3.1.1.0/24 orlonger;
            }
            then accept;
```

```
        }
        term 2 {
            then reject;
        }
    }
    policy-statement ecmp {
        term term1 {
            from protocol evpn;
            then {
                load-balance per-packet;
                accept;
            }
        }
        term term2 {
            then {
                load-balance per-packet;
                accept;
            }
        }
    }
}
routing-instances {
    VRF2_MAC_VRF_AWARE {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway no-gateway-community;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-aware;
        route-distinguisher 3.1.1.5:2122;
        vrf-target target:2122:2122;
        vlans {
            VLAN21 {
                vlan-id 21;
                l3-interface irb.21;
                vxlan {
                    vni 10021;
                }
            }
            VLAN22 {
                vlan-id 22;
                interface ae2.22;
                l3-interface irb.22;
                vxlan {
                    vni 10022;
                }
            }
        }
    }
}
routing-options {
    route-distinguisher-id 3.1.1.5;
    router-id 3.1.1.5;
    autonomous-system 65005;
    forwarding-table {
        export ecmp;
        chained-composite-next-hop {
            ingress {
                evpn;
            }
        }
    }
```

```
}
protocols {
    bgp {
        group UNDERLAY {
            type external;
            export UNDERLAY_EXPORT;
            local-as 65005;
            multipath {
                multiple-as;
            }
            neighbor 161.0.1.4 {
                description SPINE-1;
                peer-as 65011;
            }
            neighbor 161.0.1.14 {
                description SPINE-2;
                peer-as 65011;
            }
        }
        group OVERLAY {
            type external;
            multihop;
            local-address 3.1.1.5;
            family evpn {
                signaling {
                    loops 2;
                }
            }
            local-as 65005;
            multipath {
                multiple-as;
            }
            neighbor 151.0.0.11 {
                description SPINE-1;
                local-address 3.1.1.5;
                peer-as 65011;
            }
            neighbor 151.0.0.12 {
                description SPINE-2;
                local-address 3.1.1.5;
                peer-as 65011;
            }
            vpn-apply-export;
        }
    }
```

Validating Host to Host Connectivity

Our primary check is to make sure Host1 in VLAN21 can reach Host5 in VLAN22.

Initially the packet is bridged to the gateway of vlan21 on Leaf1 from Host1. The destination IP address of this ICMP-request packet is 22.0.0.15 in VLAN22. Since the IRB for VLAN22 IRB is also configured on Leaf1, it will route the packet into VLAN22 IRB locally. LEAF1 will have the TYPE2 route of Host5 existing in its database. So, LEAF1 will use this TYPE2 route and will bridge the packet to LEAF5. The L2VNI used in the VXLAN header of this packet will be 10022 (VLAN22). Once the packet reaches Leaf5, it will decapsulate it and bridge it to the destination (HOST5).

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):

```
root@Leaf-5a# run show ethernet-switching vxlan-tunnel-end-point remote
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   3.1.1.5          lo0.0    0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-IP        Flags
 3.1.1.1          VRF2_MAC_VRF_AWARE       390       vtep.32769   730     RNVE
```

```
   VNID          MC-Group-IP
   10022         0.0.0.0
   10021         0.0.0.0
 RVTEP-IP        L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-IP        Flags
 3.1.1.2         VRF2_MAC_VRF_AWARE       391       vtep.32770   727     RNVE
   VNID          MC-Group-IP
   10022         0.0.0.0
   10021         0.0.0.0
 RVTEP-IP        L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-IP        Flags
 3.1.1.3         VRF2_MAC_VRF_AWARE       392       vtep.32771   728     RNVE
   VNID          MC-Group-IP
   10022         0.0.0.0
   10021         0.0.0.0
 RVTEP-IP        L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-IP        Flags
 3.1.1.4         VRF2_MAC_VRF_AWARE       393       vtep.32772   729     RNVE
   VNID          MC-Group-IP
   10022         0.0.0.0
   10021         0.0.0.0
 RVTEP-IP        L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-IP        Flags
 3.1.1.6         VRF2_MAC_VRF_AWARE       394       vtep.32773   736     RNVE
   VNID          MC-Group-IP
   10022         0.0.0.0
   10021         0.0.0.0
 RVTEP-IP        L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-IP        Flags
 3.1.1.7         VRF2_MAC_VRF_AWARE       395       vtep.32774   737     RNVE
   VNID          MC-Group-IP
   10021         0.0.0.0
   10022         0.0.0.0

root@Leaf-5a> show interfaces irb
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 134, SNMP ifIndex: 505
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514
  Device flags   : Present Running
  Interface Specific flags: Internal: 0x200
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 2c:6b:f5:58:a0:f0, Hardware address: 2c:6b:f5:58:a0:f0
  Last flapped   : Never
    Input packets : 0
    Output packets: 0
```

Let’s track the MAC address of HOST5 2c:6b:f5:bb:f0:f0 in the below outputs.

On LEAF1 both IRB’s 21 and 22 are configured but HOST1 exists on VLAN21.

```
Leaf-1#show ip interface brief
                                                                              Address
Interface       IP Address         Status      Protocol         MTU    Owner
--------------- ------------------ ----------- ------------- ---------- -------
Vlan21          21.0.0.1/24        up          up               9200
Vlan22          22.0.0.1/24        up          up               9200


Leaf-1#show vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et4, Et5, Et6, Et7, Et8
21    VLAN21                           active    Cpu, Et3, Vx1
22    VLAN22                           active    Cpu, Vx1
```

HOST5’s MAC address is installed in the VXLAN table of LEAF1 after a TYPE2 MAC-IP route reaches for the same. We can also see the VTEP’s via which this MAC address is reachable.

```
Leaf-1#show vxlan address-table address 2c:6b:f5:bb:f0:f0
          Vxlan Mac Address Table
----------------------------------------------------------------------
```

```
VLAN  Mac Address     Type      Prt  VTEP              Moves   Last Move
----  -----------     ----      ---  ----              -----   ---------
  22  2c6b.f5bb.f0f0  EVPN      Vx1  3.1.1.5           1       1:36:05 ago
                                     3.1.1.6
Total Remote Mac Addresses for this criterion: 1
```

TYPE2 MAC+IP route for the MAC address of HOST5 on LEAF1.

```
Leaf-1#show bgp evpn route-type mac-ip 2c:6b:f5:bb:f0:f0 detail
BGP routing table information for VRF default
Router identifier 151.0.0.1, local AS number 65001
BGP routing table entry for mac-ip 10022 2c6b.f5bb.f0f0, Route Distinguisher: 3.1.1.5:2122
 Paths: 2 available
  65011 65005
    3.1.1.5 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP 
contributor
      Extended Community: Route-Target-AS:2122:2122 TunnelEncap:tunnelTypeVxlan
      Rx path id: 0x1
      VNI: 10022 ESI: 0000:0000:0000:0000:5555
  65011 65005
    3.1.1.5 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:2122:2122 TunnelEncap:tunnelTypeVxlan
      VNI: 10022 ESI: 0000:0000:0000:0000:5555
```

```
root@Leaf-5a> show ethernet-switching table 2c:6b:f5:bb:f0:f0

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


Ethernet switching table : 7 entries, 7 learned
Routing instance : VRF2_MAC_VRF_AWARE
   Vlan                MAC                 MAC       GBP    Logical                 SVLBNH/      Active
   name                address             flags     tag    interface               VENH Index   source
   VLAN22              2c:6b:f5:bb:f0:f0   DLR              ae2.22
```

Finally, the end to end connectivity using Asymmetric IRB's between HOST1 and HOST5 which are in 2 different VLANs.

```
Host1#ping 22.0.0.15 source 21.0.0.11
PING 22.0.0.15 (22.0.0.15) 72(100) bytes of data.
80 bytes from 22.0.0.15: icmp_seq=1 ttl=63 time=46.2 ms
80 bytes from 22.0.0.15: icmp_seq=2 ttl=63 time=37.5 ms
80 bytes from 22.0.0.15: icmp_seq=3 ttl=63 time=37.4 ms
80 bytes from 22.0.0.15: icmp_seq=4 ttl=63 time=39.4 ms
80 bytes from 22.0.0.15: icmp_seq=5 ttl=63 time=45.4 ms

--- 22.0.0.15 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 45ms
rtt min/avg/max/mdev = 37.458/41.235/46.275/3.860 ms, pipe 5, ipg/ewma 11.347/43.850 ms
```

Symmetric IRB on Arista's EOS and Symmetric IRB on Junos OS

Overview and Topology

Arista's EOS supports Symmetric IRB just like Junos OS. Let's look at the topology below and follow the communication between Host1 and Host6.

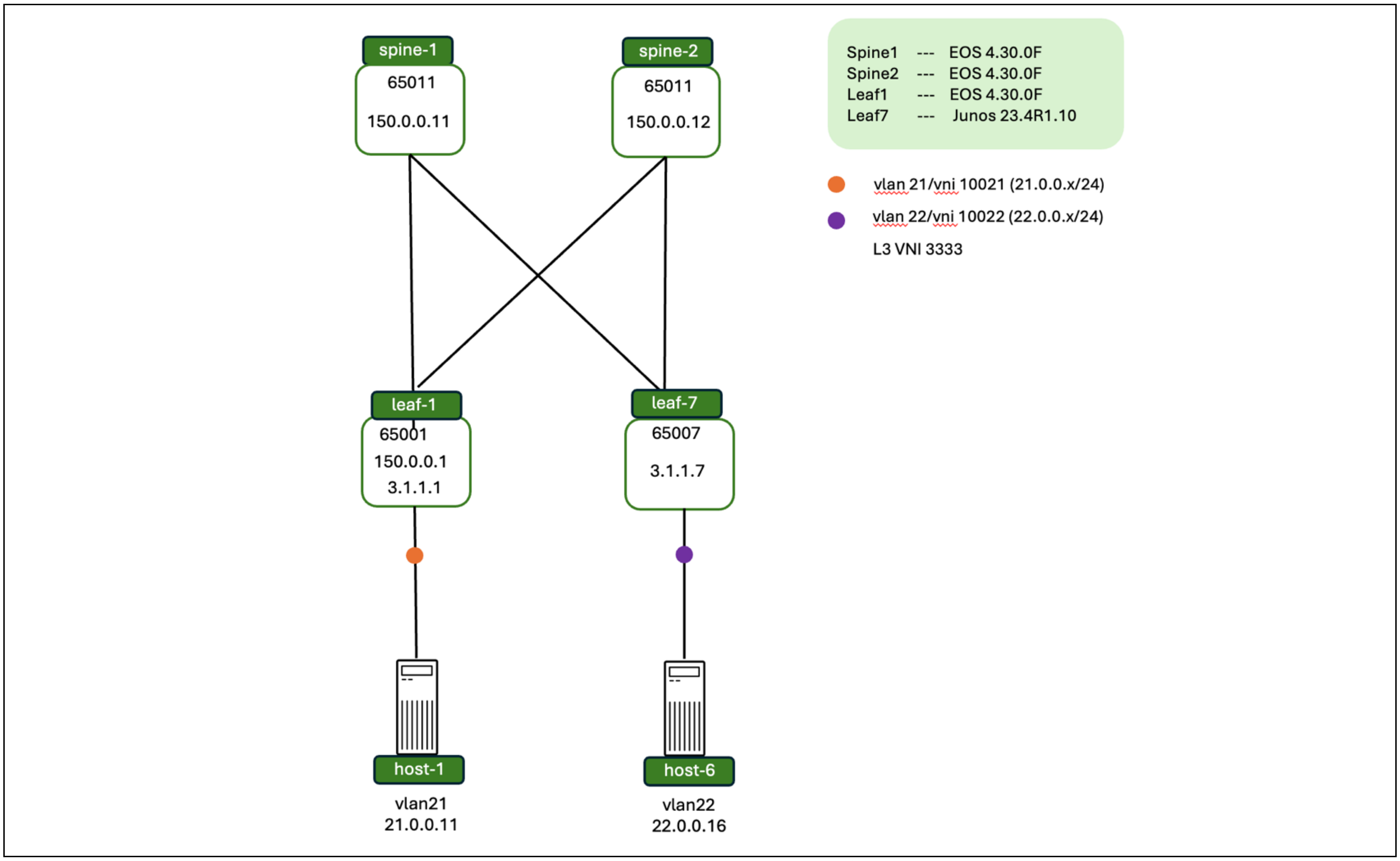

Figure 3 : Topology Diagram for Symmetric IRB on Arista's EOS and Junos OS

In a symmetric IRB design, each VTEP only knows about the subnets directly connected to its local hosts. Communication with remote subnets hosted on different VTEPs is facilitated through an intermediate routing domain—an IP-VRF shared by both VTEPs. This design enables seamless inter-subnet communication across the fabric.

The forwarding process for symmetric IRB is shown in the diagram, illustrating traffic flow from host-1 in SVI-21 (orange) to host-2 in remote SVI-22 (violet). The ingress VTEP first routes the packet from SVI-21 to the shared IP-VRF. The packet is then forwarded across the fabric to the egress VTEP, which performs a second routing operation, sending the packet from the IP-VRF to SVI-22.

Since both the ingress and egress VTEPs are responsible for routing at different stages, this approach is called symmetric IRB.

Data Plane Flow in Symmetric IRB

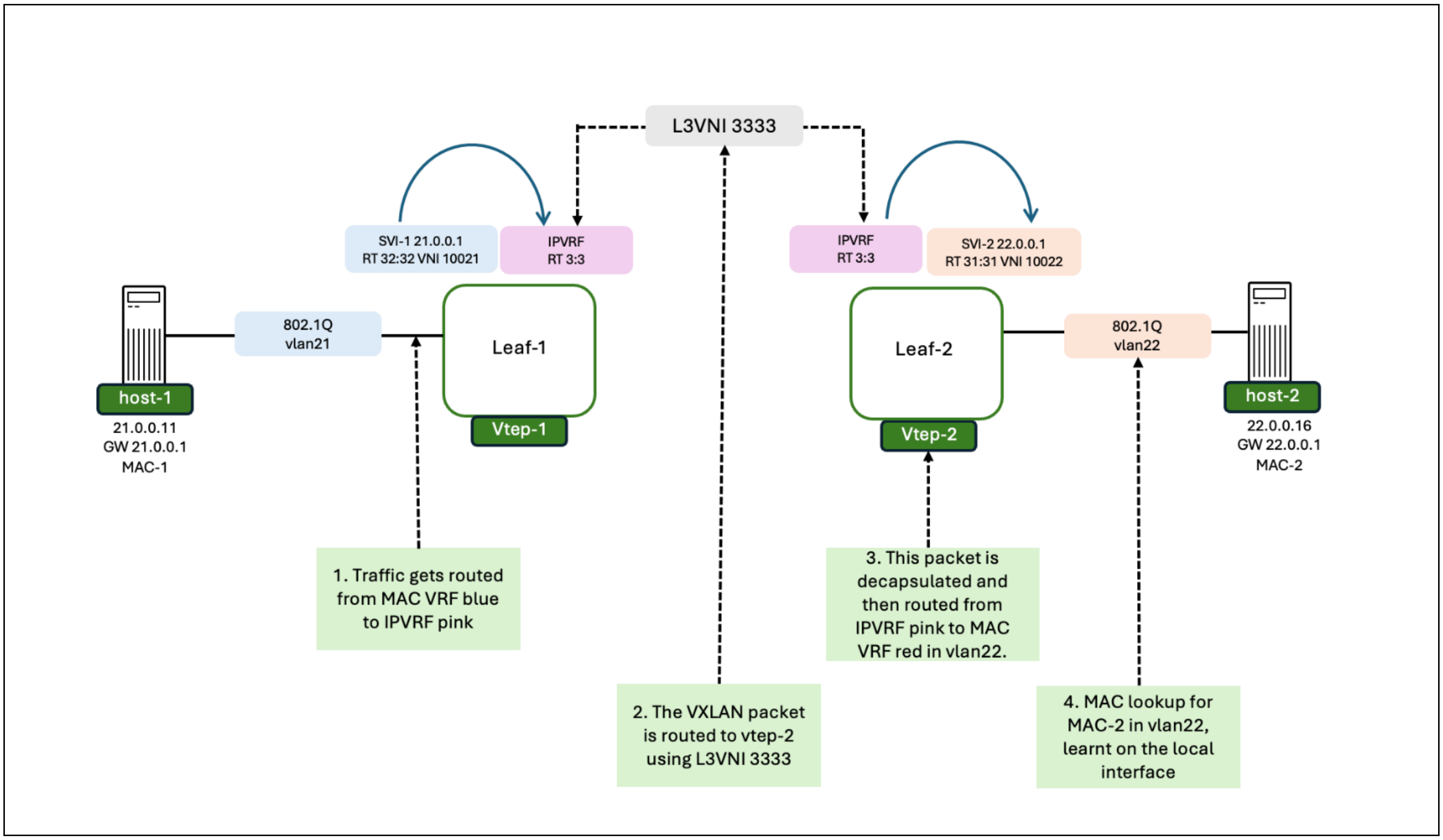

Figure 4 : Symmetric Data Plane Flow Diagram

Configuring Symmetric IRB on Arista's EOS

Host1 is in VLAN21 which is connected to Leaf1 along with its corresponding IRB's and VNI's. Since the traffic must be routed from VLAN21 (Host1) to VLAN22 (Host6) on Leaf7, an IP VRF needs to be configured on both leafs. A L3VNI of 3333 is also configured for the IP VRF. This will be used in the VXLAN header for routing.
Under the IP VRF stanza, a route distinguisher RD, a route-target RT 3:3 for importing and exporting EVPN Type5 routes is configured.

```
Leaf-1(config)#show run
*snip*

vrf instance VRF3_IP_VRF
interface Vlan21
   mtu 9200
   vrf VRF3_IP_VRF
   ip address 21.0.0.1/24
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 21 vni 10021
   vxlan vrf VRF3_IP_VRF vni 3333

ip routing vrf VRF3_IP_VRF

router bgp 65001

vlan-aware-bundle VRF3_MAC_VRF_AWARE
      rd 3.1.1.1:31
      route-target both 32:32
      redistribute learned
```

```
      vlan 21

vrf VRF3_IP_VRF
      rd 3.1.1.1:3
      route-target import evpn 3:3
      route-target export evpn 3:3
      redistribute connected
*snip*
```

Configuring Symmetric IRB on Junos OS

On Junos OS, Symmetric IRB is configured which means only VLAN22 is configured with its corresponding IRB and VNI. An IP VRF is also configured similar to Arista's EOS with IRB 22. The IP VRF also has a RD and RT of 3:3. A L3VNI of 3333 is also needed to be configured here for VXLAN routing.

```
root@Leaf-7> show configuration
*snip*
irb {
        unit 22 {
            family inet {
                address 22.0.0.1/24;
            }
        }
    }
routing-instances {
    VRF3_IP_VRF {
        instance-type vrf;
        routing-options {
            multipath;
        }
        protocols {
            evpn {
                irb-symmetric-routing {
                vni 3333;
            }
                ip-prefix-routes {
                    advertise direct-nexthop;
                    encapsulation vxlan;
                    vni 3333;
                    export s1;
                }
            }
        }
        interface irb.22;
        interface lo0.2;
        route-distinguisher 3.1.1.7:3;
        vrf-target target:3:3;
    }
    VRF3_MAC_VRF_AWARE {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway no-gateway-community;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-aware;
        route-distinguisher 3.1.1.7:22;
        vrf-target target:31:31;
        vlans {
            VLAN22 {
                vlan-id 22;
                interface ge-0/0/2.22;
                l3-interface irb.22;
                vxlan {
                    vni 10022;
                }
            }
```

```
        }
    }
}
```

Validating Host to Host Connectivity

Here, let's check the connectivity between Host1 (VLAN21) and Host6 (VLAN22). As soon as an IP VRF is configured with same L3VNI 3333, a TYPE5 route is generated and sent to all leafs, but it will only be accepted and installed if the leaf has a similar RT configured under the IP VRF.
Leaf7 installs this Type5 route as it has an IP VRF with same RT 3:3 and vice versa.
When Host1 tries to send an ICMP packet to Host6, the destination IP 22.0.0.16 is in a different subnet. So, Leaf1 routes the packet to Leaf7 based on the available Type5 route. Once the packet reaches Leaf7, it decapsulates the packet and since the destination IP is in VLAN22, the packet will again be routed to IRB22 and bridged to Host6.

Once the MAC address of Host6 is learned on Leaf7, it is installed in the Ethernet-table and the EVPN table for that MAC VRF.

```
root@Leaf-7> show ethernet-switching table

Ethernet switching table : 1 entries, 1 learned
Routing instance : VRF3_MAC_VRF_AWARE
   Vlan                MAC                 MAC       GBP    Logical               SVLBNH/      Active
   name                address             flags     tag    interface             VENH Index   source
   VLAN22              2c:6b:f5:c1:2d:f0   D                ge-0/0/2.22


root@Leaf-7> show route table VRF3_MAC_VRF_AWARE.evpn.0

VRF3_MAC_VRF_AWARE.evpn.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:3.1.1.7:22::10022::2c:6b:f5:20:e5:f0/304 MAC/IP
                   *[EVPN/170] 21:42:53
                       Indirect
2:3.1.1.7:22::10022::2c:6b:f5:c1:2d:f0/304 MAC/IP
                   *[EVPN/170] 21:42:47
                       Indirect
2:3.1.1.7:22::10022::2c:6b:f5:20:e5:f0::22.0.0.1/304 MAC/IP
                   *[EVPN/170] 00:33:00
                       Indirect
2:3.1.1.7:22::10022::2c:6b:f5:c1:2d:f0::22.0.0.16/304 MAC/IP
                   *[EVPN/170] 00:33:00
                       Indirect
3:3.1.1.7:22::10022::3.1.1.7/248 IM
                   *[EVPN/170] 21:42:52
                       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@Leaf-7> show route table VRF3_MAC_VRF_AWARE.evpn.0 evpn-mac-address 2c:6b:f5:c1:2d:f0 extensive

VRF3_MAC_VRF_AWARE.evpn.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
2:3.1.1.7:22::10022::2c:6b:f5:c1:2d:f0/304 MAC/IP (1 entry, 1 announced)
        *EVPN   Preference: 170
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7ca6114
                Next-hop reference count: 13
                Kernel Table Id: 0
                Protocol next hop: 3.1.1.7
                Indirect next hop: 0x0 - INH Session ID: 0
                Indirect next hop: INH non-key opaque: 0x0 INH key opaque: 0x0
                State: <Active Int Ext>
                Age: 21:48:55
                Validation State: unverified
                Task: VRF3_MAC_VRF_AWARE-evpn
                Announcement bits (1): 2-rt-export
                AS path: I
                Communities: encapsulation:vxlan(0x8)
```

```
                Route Label: 10022
                ESI: 00:00:00:00:00:00:00:00:00:00
                Thread: junos-main

2:3.1.1.7:22::10022::2c:6b:f5:c1:2d:f0::22.0.0.16/304 MAC/IP (1 entry, 1 announced)
        *EVPN   Preference: 170
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7ca6114
                Next-hop reference count: 13
                Kernel Table Id: 0
                Protocol next hop: 3.1.1.7
                Indirect next hop: 0x0 - INH Session ID: 0
                Indirect next hop: INH non-key opaque: 0x0 INH key opaque: 0x0
                State: <Active Int Ext>
                Age: 39:08
                Validation State: unverified
                Task: VRF3_MAC_VRF_AWARE-evpn
                Announcement bits (1): 2-rt-export
                AS path: I
                Communities: target:3:3 encapsulation:vxlan(0x8) router-mac:2c:6b:f5:20:e5:f0
                Route Label: 10022
                Route Label: 3333
                ESI: 00:00:00:00:00:00:00:00:00:00
                Thread: junos-main
```

Leaf7 receives a TYPE-5 route from Leaf1 for 21.0.0.0/24 subnet.

```
root@Leaf-7> show route table VRF3_IP_VRF.evpn.0

VRF3_IP_VRF.evpn.0: 4 destinations, 5 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

5:3.1.1.1:3::0::21.0.0.0::24/248
                   *[BGP/170] 00:12:56, localpref 100, from 151.0.0.11
                      AS path: 65011 65001 I, validation-state: unverified
                       to 161.0.1.18 via ge-0/0/0.0, Push 3333
                    >  to 161.0.1.26 via ge-0/0/1.0, Push 3333
                    [BGP/170] 00:12:55, localpref 100, from 151.0.0.12
                      AS path: 65011 65001 I, validation-state: unverified
                       to 161.0.1.18 via ge-0/0/0.0, Push 3333
                    >  to 161.0.1.26 via ge-0/0/1.0, Push 3333
```

Leaf1 receives a TYPE-5 route from Leaf7 for 22.0.0.0/24 subnet as well as 22.0.0.16/32 host route.

```
Leaf-1#show bgp evpn route-type ip-prefix 22.0.0.16/32
BGP routing table information for VRF default
Router identifier 151.0.0.1, local AS number 65001
BGP routing table entry for ip-prefix 22.0.0.16/32, Route Distinguisher: 3.1.1.7:3
 Paths: 2 available
  65011 65007
    3.1.1.7 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP
contributor
      Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:2c:6b:f5:20:e5:f0
      Rx path id: 0x1
      VNI: 3333
  65011 65007
    3.1.1.7 from 151.0.0.12 (151.0.0.12)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
      Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:2c:6b:f5:20:e5:f0
      VNI: 3333


Leaf-1#show ip route vrf VRF3_IP_VRF 22.0.0.16/32

VRF: VRF3_IP_VRF
 B E      22.0.0.16/32 [20/0] via VTEP 3.1.1.7 VNI 3333 router-mac 2c:6b:f5:20:e5:f0 local-interface Vxlan1
```

When Leaf1 receives icmp-request packet from host1 with destination IP as 22.0.0.16, it uses this TYPE-5 route to send the packet to Leaf7 using L3VNI 3333.

Here is the packet capture on Spine1 of the same icmp-request packet.

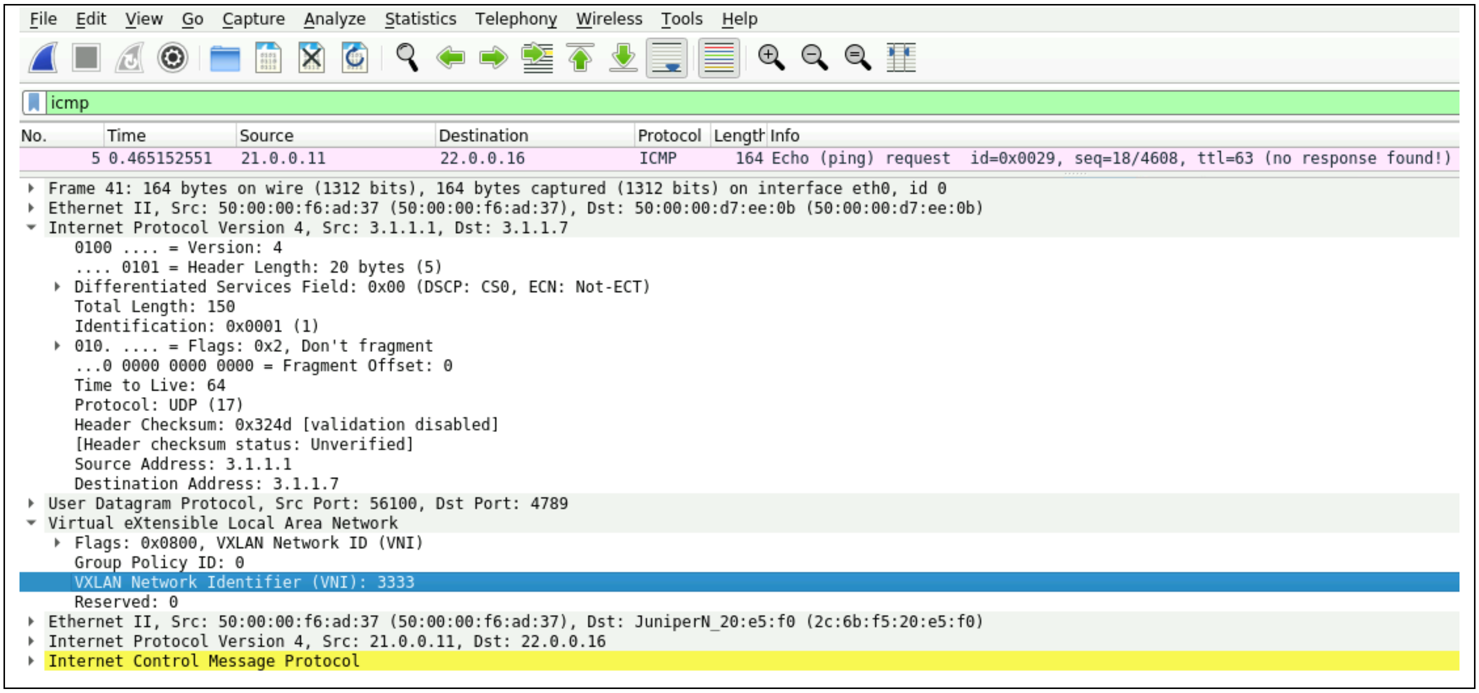

Figure 5 : Packet capture of an ICMP request packet on Spine1 using L3VNI 3333

Pings between Host1 and Host6.

```
Host1#ping 22.0.0.16
PING 22.0.0.16 (22.0.0.16) 72(100) bytes of data.
80 bytes from 22.0.0.16: icmp_seq=1 ttl=62 time=29.0 ms
80 bytes from 22.0.0.16: icmp_seq=2 ttl=62 time=30.7 ms
80 bytes from 22.0.0.16: icmp_seq=3 ttl=62 time=22.4 ms
80 bytes from 22.0.0.16: icmp_seq=4 ttl=62 time=26.1 ms
80 bytes from 22.0.0.16: icmp_seq=5 ttl=62 time=23.8 ms

--- 22.0.0.16 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 81ms
rtt min/avg/max/mdev = 22.488/26.452/30.772/3.097 ms, pipe 3, ipg/ewma 20.301/27.580 ms
```

DCI Between Arista's EOS and Junos OS Fabrics for Bridged Overlay

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.

Overview and Topology

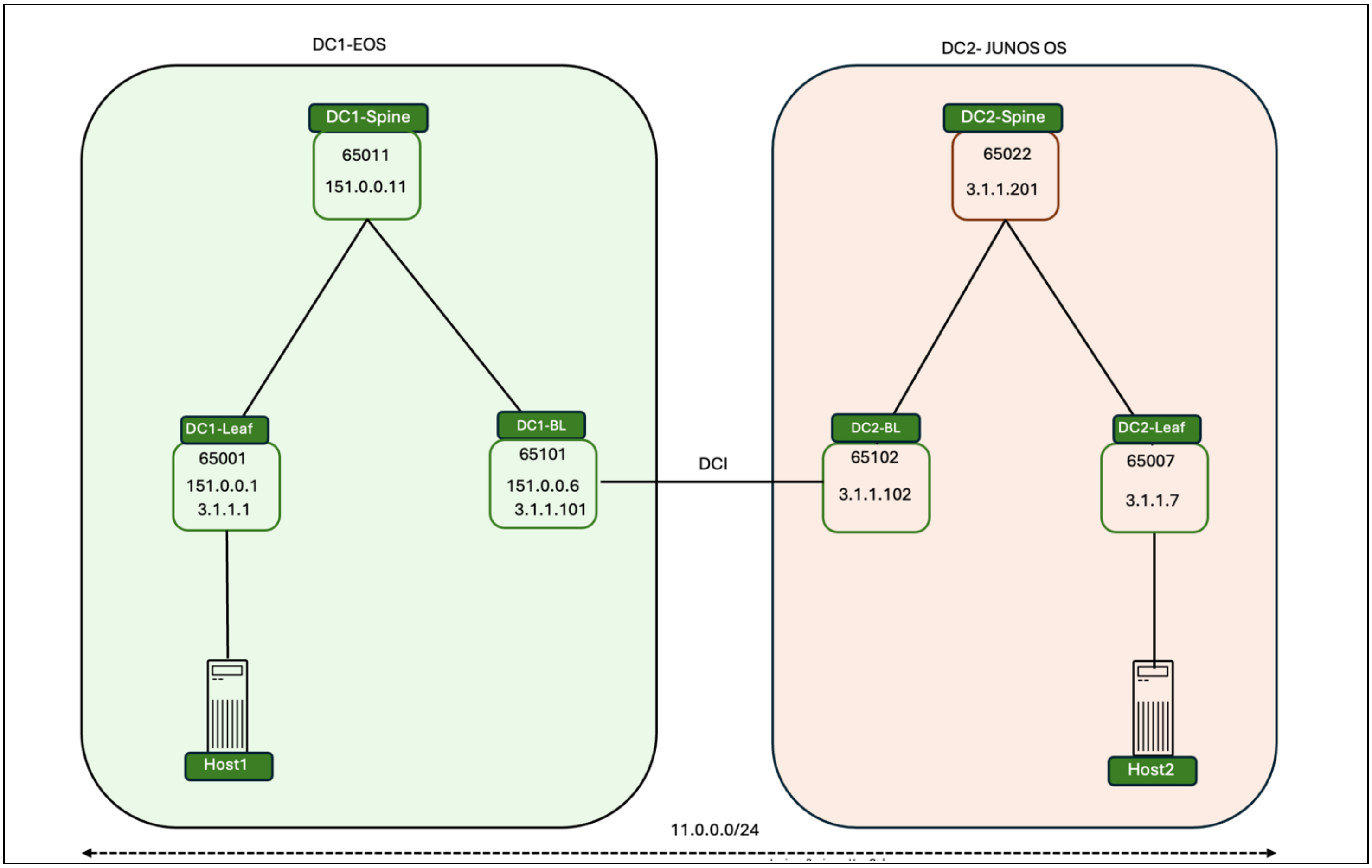

Figure 6 : Topology of 2 Vendor Fabrics Connected Via DCI

As seen above, DC1 comprises of Arista's EOS 3-stage CLOS architecture with DC1-BL 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-BL. Both hosts Host1 and Host2 are in the same VLAN11 and subnet 11.0.0.0/24. The goal is to stretch the bridges overlay across DC's and ensure that both hosts can reach each other.

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

Arista's EOS supports:

- **RFC 9014** which is the gateway DCI solution with multiple control planes (VPLS/EVPN) and data-planes (MPLS, VXLAN, PBB).
- **draft-sharma-bess-multi-site-evpn-01** which is the GW DCI solution focused only on EVPN-VXLAN, support for a single control plane (EVPN) and single data plane (VXLAN).
- **draft-ietf-bess-evpn-ipvpn-interworking-07** which is the Layer3 DCI interop between EVPN-VXLAN/MPLS and IP-VPN WAN for layer 3 DCI

Junos OS and Junos OS Evolved implements 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 Gateway Interconnect on Arista's EOS

The spine and leaf configurations are the same as the bridged overlay section. We'll post the configuration from the Arista's EOS spine and leaf.
A snippet from DC1-Spine1.

```
DC1-Spine1#show run
*snip*
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.11/32
!
ip routing
!
ip prefix-list PL-loopback-1 seq 10 permit 3.1.1.0/24 eq 32
ip prefix-list PL-loopback-evpn seq 20 permit 151.0.0.0/24 eq 32
!
ip route 0.0.0.0/0 10.92.71.254
ip route 0.0.0.0/0 10.102.197.254
!
route-map RM-BGP permit 10
   match ip address prefix-list PL-loopback-evpn
!
route-map RM-BGP-1 permit 10
   match ip address prefix-list PL-loopback-1
!
router bgp 65011
   router-id 151.0.0.11
   no bgp default ipv4-unicast
   distance bgp 20 200 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY next-hop-unchanged
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor IPv4-UNDERLAY peer group
   neighbor IPv4-UNDERLAY send-community
   neighbor IPv4-UNDERLAY maximum-routes 9000
   neighbor 151.0.0.1 peer group EVPN-OVERLAY
   neighbor 151.0.0.1 remote-as 65001
   neighbor 151.0.0.1 description Leaf-1
   neighbor 151.0.0.6 peer group EVPN-OVERLAY
   neighbor 151.0.0.6 remote-as 65101
   neighbor 151.0.0.6 description DC1-BL
   neighbor 161.0.1.5 peer group IPv4-UNDERLAY
   neighbor 161.0.1.5 remote-as 65101
   neighbor 161.0.1.5 description DC1-BL
   neighbor 161.0.1.21 peer group IPv4-UNDERLAY
   neighbor 161.0.1.21 remote-as 65001
   neighbor 161.0.1.21 description Leaf-1
   redistribute connected route-map RM-BGP
   !
   address-family evpn
      bgp additional-paths send ecmp
      neighbor EVPN-OVERLAY activate
   !
   address-family ipv4
      no neighbor EVPN-OVERLAY activate
      neighbor IPv4-UNDERLAY activate
```

A snippet from DC1-Leaf1.

```
vlan 11
   name VLAN11
!
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.1/32
!
interface Loopback1
   description VTEP_vxlan
   ip address 3.1.1.1/32
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 11 vni 10011
!
mac address-table aging-time 21600
!
ip virtual-router mac-address 00:00:00:ab:ab:01
!
ip routing
!
ip prefix-list PL-loopback-overlay
   seq 10 permit 151.0.0.0/24 eq 32
   seq 20 permit 3.1.1.0/24 eq 32
!
route-map RM-eBGP permit 10
   match ip address prefix-list PL-loopback-overlay
!
router bgp 65001
   router-id 3.1.1.1
   no bgp default ipv4-unicast
   distance bgp 20 100 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor IPv4-UNDERLAY peer group
   neighbor IPv4-UNDERLAY send-community
   neighbor IPv4-UNDERLAY maximum-routes 9000
   neighbor 151.0.0.11 peer group EVPN-OVERLAY
   neighbor 151.0.0.11 remote-as 65011
   neighbor 151.0.0.11 description Spine-1
   neighbor 161.0.1.20 peer group IPv4-UNDERLAY
   neighbor 161.0.1.20 remote-as 65011
   neighbor 161.0.1.20 description Spine-1
   redistribute connected route-map RM-eBGP
   !
   vlan 11
      rd 3.1.1.1:11
      route-target both 11:11
      redistribute learned
   !
   address-family evpn
      neighbor EVPN-OVERLAY activate
   !
   address-family ipv4
      no neighbor EVPN-OVERLAY activate
      neighbor IPv4-UNDERLAY activate
   !
!
```

On DC1-Leaf1, vlan 11 is configured and vni 10011 is associated with it. The route-target 11:11 is used for local EVPN Domain. Same RT will be configured on DC1-BL as well. Let’s look at the config:

A snippet from DC1-BL.

```
vlan 11
   name VLAN11
!
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.6/32
!
interface Loopback1
   description VTEP_vxlan
   ip address 3.1.1.101/32
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 11 vni 10011
!
ip prefix-list PL-loopback-overlay
   seq 10 permit 151.0.0.0/24 eq 32
   seq 20 permit 3.1.1.0/24 eq 32
!
route-map RM-eBGP permit 10
   match ip address prefix-list PL-loopback-overlay
!
router bgp 65101
   router-id 151.0.0.6
   no bgp default ipv4-unicast
   distance bgp 20 100 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor EVPN-OVERLAY-dci peer group
   neighbor EVPN-OVERLAY-dci update-source Loopback0
   neighbor EVPN-OVERLAY-dci bfd
   neighbor EVPN-OVERLAY-dci ebgp-multihop 3
   neighbor EVPN-OVERLAY-dci send-community
   neighbor EVPN-OVERLAY-dci maximum-routes 0
   neighbor IPv4-UNDERLAY peer group
   neighbor IPv4-UNDERLAY send-community
   neighbor IPv4-UNDERLAY maximum-routes 9000
   neighbor 3.1.1.102 peer group EVPN-OVERLAY-dci
   neighbor 3.1.1.102 remote-as 65102
   neighbor 3.1.1.102 description DC2-BL
   neighbor 151.0.0.11 peer group EVPN-OVERLAY
   neighbor 151.0.0.11 remote-as 65011
   neighbor 151.0.0.11 description Spine-1
   neighbor 161.0.1.4 peer group IPv4-UNDERLAY
   neighbor 161.0.1.4 remote-as 65011
   neighbor 161.0.1.4 description Spine-1
   neighbor 171.0.1.151 peer group IPv4-UNDERLAY
   neighbor 171.0.1.151 remote-as 65102
   neighbor 171.0.1.151 description DC2-BL
   redistribute connected route-map RM-eBGP
   !
   vlan 11
      rd 151.0.0.6:11his
      rd evpn domain remote 151.0.0.6:11
      route-target both 11:11
      route-target import export evpn domain remote 100:100
      redistribute learned
   !
   address-family evpn
      neighbor EVPN-OVERLAY activate
      neighbor EVPN-OVERLAY-dci activate
      neighbor EVPN-OVERLAY-dci domain remote
      neighbor default next-hop-self received-evpn-routes route-type ip-prefix inter-domain
   !
   address-family ipv4
      no neighbor EVPN-OVERLAY activate
      neighbor IPv4-UNDERLAY activate
```

The route target 11:11 is configured under vlan 11 and is used for local EVPN Domain. An additional eBGP overlay session is configured between the 2 DCI Border Leafs under the peer group "EVPN-Overlay-dci" and the neighbors under this peer group will be considered as remote DCI peers (Junos OS).
The shared route-target between both the DCI BL's is configured as 100:100.
To enable the advertisement of routes between EVPN domain from local (DC1-BL) to remote (DC2-BL) "neighbor default next-hop-self received-evpn-routes route-type ip-prefix inter-domain" is needed as well.

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-BL.
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-BL1# show routing-instances VRF1_MAC_VRF_BASED
instance-type mac-vrf;
protocols {
    evpn {
        encapsulation vxlan;
        default-gateway no-gateway-community;
        extended-vni-list all;
        interconnect {
            vrf-target target:100:100;
            route-distinguisher 3.1.1.102:100;
            esi {
                00:00:00:00:00:00:00:00:00:22;
                all-active;
            }
            interconnected-vni-list 10011;
            encapsulation vxlan;
        }
    }
}
vtep-source-interface lo0.0;
service-type vlan-based;
route-distinguisher 3.1.1.102:11;
vrf-target target:110:110;
vlans {
    VLAN11 {
        vlan-id 11;
        vxlan {
            vni 10011;
        }
    }
}
```

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 3.1.1.102:100, 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.

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.

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.

I-ESI is supported in Arista's EOS starting from EOS 4.33.1F. For this document I-ESI has not been used as it is only supported on R3 series switches whereas this topology uses a 7500R series switch as a DCI gateway. The current DCI config on Arista's EOS is the equivalent of DCI stitching without I-ESI in Junos OS terminology.

Visually, this is what it looks like:

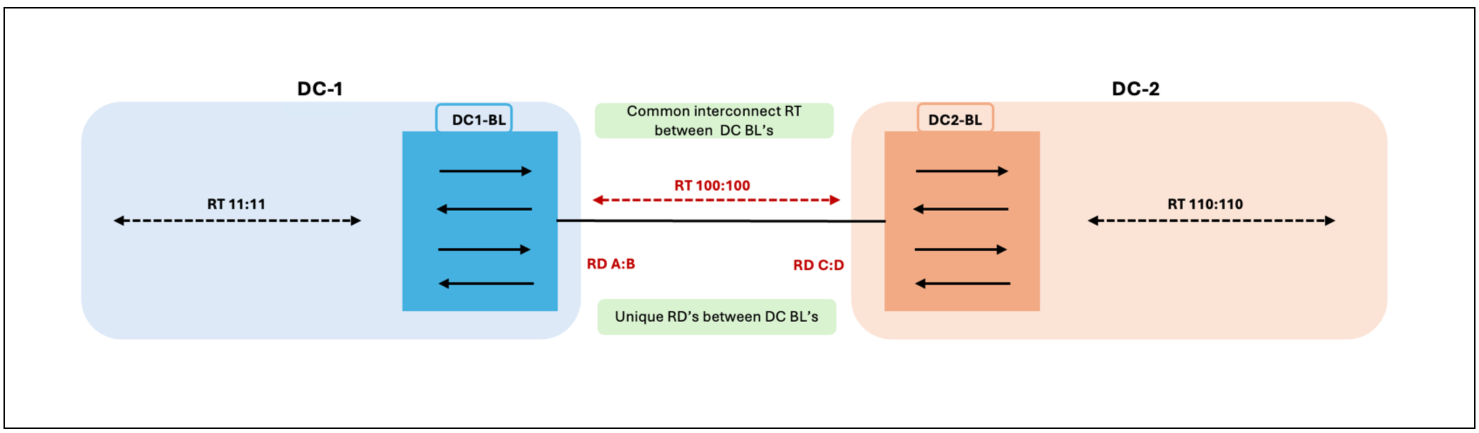

Figure 7 : Route-Targets Used Across DCs

Following Control Plane EVPN Updates from DC2 to DC1

Let's trace the flow of Host2's MAC address from DC2 to DC1 from a control plane EVPN update perspective. First, on DC2-Leaf1, Host2's MAC address is a local learn.

```
root@DC2-Leaf1> show ethernet-switching table vlan-id 11

Ethernet switching table : 1 entries, 1 learned
Routing instance : VRF1_MAC_VRF_BASED
   Vlan                 MAC                 MAC      GBP    Logical                 SVLBNH/      Active
   name                 address             flags    tag    interface               VENH Index   source
   VLAN11               c0:03:80:1c:d8:e0   D               et-0/0/2.0
```

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-Leaf1.

```
{master:0}
root@DC2-Spine1> show route table bgp.evpn.0 evpn-mac-address c0:03:80:1c:d8:e0

bgp.evpn.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both

2:3.1.1.7:11::0::c0:03:80:1c:d8:e0/304 MAC/IP
                   *[BGP/170] 00:03:46, localpref 100, from 3.1.1.7
                      AS path: 65007 I, validation-state: unverified
                    >  to 171.0.1.19 via et-0/0/49.0
2:3.1.1.102:100::0::c0:03:80:1c:d8:e0/304 MAC/IP
                   *[BGP/170] 00:03:45, localpref 100, from 3.1.1.102
                      AS path: 65102 I, validation-state: unverified
                    >  to 171.0.1.21 via et-0/0/50.0
```

More importantly, the route-target is 110:110, which is the local route-target and the route-distinguisher is of DC2-Leaf1, as expected.

```
{master:0}
root@DC2-Spine1> show route receive-protocol bgp 3.1.1.7 evpn-mac-address c0:03:80:1c:d8:e0 extensive
bgp.evpn.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
* 2:3.1.1.7:11::0::c0:03:80:1c:d8:e0/304 MAC/IP (1 entry, 1 announced)
     Accepted
     Route Distinguisher: 3.1.1.7:11
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 3.1.1.7
     AS path: 65007 I
     Communities: target:110:110 encapsulation:vxlan(0x8)

* 2:3.1.1.7:11::0::c0:03:80:1c:d8:e0::11.0.0.16/304 MAC/IP (1 entry, 1 announced)
     Accepted
     Route Distinguisher: 3.1.1.7:11
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 3.1.1.7
     AS path: 65007 I
     Communities: target:110:110 encapsulation:vxlan(0x8)
```

The spines will send this to the other leaf switches, including DC2-BL. 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-BL.

```
root@DC2-BL1> show route receive-protocol bgp 3.1.1.201 evpn-mac-address c0:03:80:1c:d8:e0 table bgp.evpn.0
extensive

bgp.evpn.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
* 2:3.1.1.7:11::0::c0:03:80:1c:d8:e0/304 MAC/IP (1 entry, 1 announced)
     Import Accepted
     Route Distinguisher: 3.1.1.7:11
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 3.1.1.7
     AS path: 65022 65007 I
     Communities: target:110:110 encapsulation:vxlan(0x8)

* 2:3.1.1.7:11::0::c0:03:80:1c:d8:e0::11.0.0.16/304 MAC/IP (1 entry, 1 announced)
     Import Accepted
     Route Distinguisher: 3.1.1.7:11
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: 3.1.1.7
     AS path: 65022 65007 I
     Communities: target:110:110 encapsulation:vxlan(0x8)
```

Below we can see border leaf DC2-BL send this update to DC1-BL.

```
root@DC2-BL1> show route advertising-protocol bgp 151.0.0.6 table bgp.evpn.0 evpn-mac-address c0:03:80:1c:d8:e0

bgp.evpn.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
```

```
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:3.1.1.7:11::0::c0:03:80:1c:d8:e0/304 MAC/IP
*                         Self                                    65022 65007 I
  2:3.1.1.102:100::0::c0:03:80:1c:d8:e0/304 MAC/IP
*                         Self                                    I
  2:3.1.1.7:11::0::c0:03:80:1c:d8:e0::11.0.0.16/304 MAC/IP
*                         Self                                    65022 65007 I
  2:3.1.1.102:100::0::c0:03:80:1c:d8:e0::11.0.0.16/304 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 00:00:00:00:00:00:00:00:00:22 attached to the route.

```
root@DC2-BL1> show route advertising-protocol bgp 151.0.0.6 table bgp.evpn.0 evpn-mac-address c0:03:80:1c:d8:e0
extensive

bgp.evpn.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
* 2:3.1.1.7:11::0::c0:03:80:1c:d8:e0/304 MAC/IP (1 entry, 1 announced)
 BGP group OVERLAY type External
     Route Distinguisher: 3.1.1.7:11
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: Self
     AS path: [65102] 65022 65007 I
     Communities: target:110:110 encapsulation:vxlan(0x8)

* 2:3.1.1.102:100::0::c0:03:80:1c:d8:e0/304 MAC/IP (1 entry, 1 announced)
 BGP group OVERLAY type External
     Route Distinguisher: 3.1.1.102:100
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:22
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65102] I
     Communities: target:100:100 encapsulation:vxlan(0x8)

* 2:3.1.1.7:11::0::c0:03:80:1c:d8:e0::11.0.0.16/304 MAC/IP (1 entry, 1 announced)
 BGP group OVERLAY type External
     Route Distinguisher: 3.1.1.7:11
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:00
     Nexthop: Self
     AS path: [65102] 65022 65007 I
     Communities: target:110:110 encapsulation:vxlan(0x8)

* 2:3.1.1.102:100::0::c0:03:80:1c:d8:e0::11.0.0.16/304 MAC/IP (1 entry, 1 announced)
 BGP group OVERLAY type External
     Route Distinguisher: 3.1.1.102:100
     Route Label: 10011
     ESI: 00:00:00:00:00:00:00:00:00:22
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65102] 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-BL.

```
DC1-BL1(s1)(config)#router bgp 65101
DC1-BL1(s1)(config-router-bgp)#show active
*snip*
   vlan 11
      rd 151.0.0.6:11
      rd evpn domain remote 151.0.0.6:11
      route-target both 11:11
      route-target import export evpn domain remote 100:100
      redistribute learned
```

This route is now accepted into BGP RIB on DC1-BL.

```
DC1-BL1(s1)#show bgp evpn route-type mac-ip c0:03:80:1c:d8:e0 detail
BGP routing table information for VRF default
Router identifier 151.0.0.6, local AS number 65101
BGP routing table entry for mac-ip c003.801c.d8e0, Route Distinguisher: 151.0.0.6:11
 Paths: 1 available
  65102
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:11:11 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip c003.801c.d8e0 11.0.0.16, Route Distinguisher: 151.0.0.6:11
 Paths: 1 available
  65102
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:11:11 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip c003.801c.d8e0 remote, Route Distinguisher: 3.1.1.102:100
 Paths: 1 available
  65102
    3.1.1.102 from 3.1.1.102 (3.1.1.102)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:100:100 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:0022
BGP routing table entry for mac-ip c003.801c.d8e0 11.0.0.16 remote, Route Distinguisher: 3.1.1.102:100
 Paths: 1 available
  65102
    3.1.1.102 from 3.1.1.102 (3.1.1.102)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:100:100 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:0022
```

It is installed in VXLAN address-table and eventually the MAC address table.

```
DC1-BL1(s1)#show vxlan address-table address  c003.801c.d8e0
          Vxlan Mac Address Table
----------------------------------------------------------------------

VLAN  Mac Address     Type      Prt  VTEP             Moves   Last Move
----  -----------     ----      ---  ----             -----   ---------
  11  c003.801c.d8e0  EVPN      Vx1  3.1.1.102        1       0:14:05 ago
Total Remote Mac Addresses for this criterion: 1


DC1-BL1(s1)# show mac address-table address c003.801c.d8e0
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
  11    c003.801c.d8e0    DYNAMIC     Vx1        1       0:14:08 ago
Total Mac Addresses for this criterion: 1
```

This is sent as a BGP EVPN update to the spines, and in turn, DC1-Leaf1. While advertising remote routes, GW is expected to regenerate the routes using its own route-distinguisher and route-target. Here the local RT on Arista's EOS side is 11:11.

```
DC1-Spine1#show bgp evpn route-type mac-ip c0:03:80:1c:d8:e0 detail
BGP routing table information for VRF default
Router identifier 151.0.0.11, local AS number 65011
BGP routing table entry for mac-ip c003.801c.d8e0, Route Distinguisher: 151.0.0.6:11
 Paths: 1 available
```

```
  65101 65102
    3.1.1.101 from 151.0.0.6 (151.0.0.6)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:11:11 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip c003.801c.d8e0 11.0.0.16, Route Distinguisher: 151.0.0.6:11
 Paths: 1 available
  65101 65102
    3.1.1.101 from 151.0.0.6 (151.0.0.6)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:11:11 TunnelEncap:tunnelTypeVxlan
      VNI: 10011 ESI: 0000:0000:0000:0000:0000
```

The update is reflected to the leaf, and dc1-leaf1 now installs this in the MAC address table.

```
DC1-Leaf1#show vxlan address-table address c0:03:80:1c:d8:e0
          Vxlan Mac Address Table
----------------------------------------------------------------------

VLAN  Mac Address     Type      Prt  VTEP             Moves   Last Move
----  -----------     ----      ---  ----             -----   ---------
  11  c003.801c.d8e0  EVPN      Vx1  3.1.1.101        1       0:22:23 ago
Total Remote Mac Addresses for this criterion: 1

DC1-Leaf1#show mac address-table address c0:03:80:1c:d8:e0
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
  11    c003.801c.d8e0    DYNAMIC     Vx1        1       0:22:25 ago
Total Mac Addresses for this criterion: 1
```

This concludes the control plane updates from DC2 to DC1.

Verify DC1 Host1's MAC Address is Learnt in DC2-Leaf1 as Remote MAC

On DC1-Leaf1, Host1's MAC address c003.801d.a0e0 is a local learn.

```
DC1-Leaf1#show mac address-table address c003.801d.a0e0
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
  11    c003.801d.a0e0    DYNAMIC     Et6        1       4 days, 1:07:22 ago
Total Mac Addresses for this criterion: 1
```

On DC2-Leaf1, DC1-Host1's MAC address c003.801d.a0e0 is learnt remotely. I-ESI is also attached to this mac-address. This Type-2 route is received on DC2-BL from DC1-BL and DC2-BL readvertises the same route to DC2-Leaf1.

```
root@DC2-Leaf1> show ethernet-switching table vlan-id 11 c0:03:80:1d:a0:e0

Ethernet switching table : 2 entries, 2 learned
Routing instance : VRF1_MAC_VRF_BASED
   Vlan                MAC                 MAC       GBP    Logical                SVLBNH/      Active
   name                address             flags     tag    interface              VENH Index   source
   VLAN11              c0:03:80:1d:a0:e0   DR               esi.83789
00:00:00:00:00:00:00:00:00:22
```

Ping from DC2-Host2 to DC1-Host1.

```
root@DC2-Host2> ping 11.0.0.11 count 5
```

```
PING 11.0.0.11 (11.0.0.11): 56 data bytes
64 bytes from 11.0.0.11: icmp_seq=0 ttl=64 time=61.372 ms
64 bytes from 11.0.0.11: icmp_seq=1 ttl=64 time=15.033 ms
64 bytes from 11.0.0.11: icmp_seq=2 ttl=64 time=9.826 ms
64 bytes from 11.0.0.11: icmp_seq=3 ttl=64 time=10.694 ms
64 bytes from 11.0.0.11: icmp_seq=4 ttl=64 time=9.263 ms

--- 11.0.0.11 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.263/21.238/61.372/20.170 ms
```

DCI Between Arista's EOS and Junos OS Fabrics Using T5 Stitching

Overview and Topology

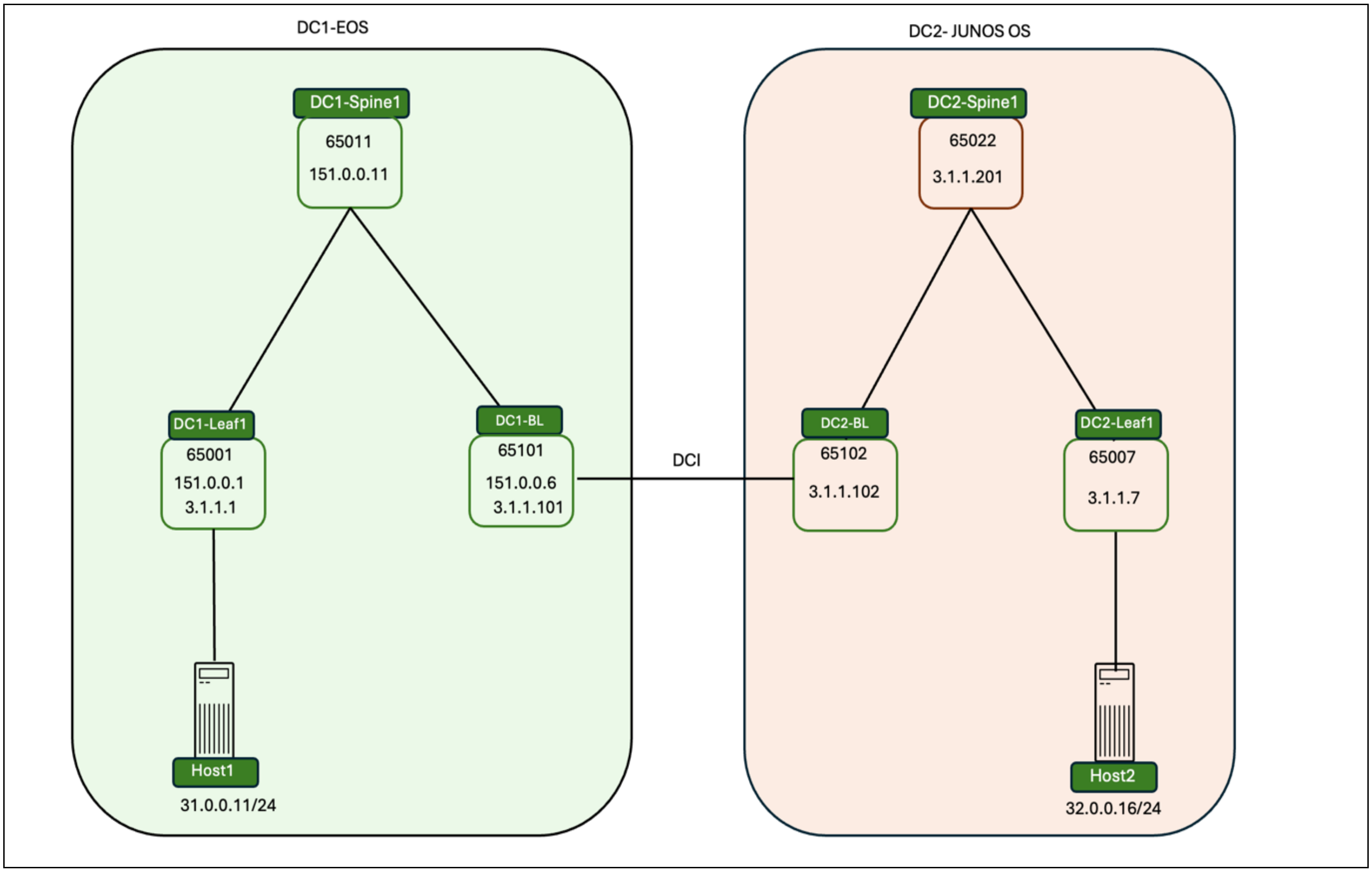

Figure 8 : Topology of 2 Vendor Fabrics Connected Via DCI with Hosts in Different VLANs

The previous section was stretching L2 across 2 data centers but there might be times when some of the tenants are in different VLANs and IP prefixes i.e, they are DC site specific. Sometimes only Type-5 routes are advertised within the DC which helps in scaling out the fabric instead of MAC and MAC/IP advertisements.
There might be situations where Type-5 and Type-2 routes are advertised inside a DC fabric, but only Type-5 routes are advertised for DCI purposes to have full reachability.

In this section hosts Host1 and Host2 are in different VLANs/subnet VLAN31 in DC1 and VLAN32 in DC2 with respective subnets 31.0.0.0/24 and 32.0.0.0/24. The goal is to ensure that both hosts can reach each other using Type5 routes.

Configuring Arista's EOS Leaf1

In this scenario Data centers maintains separate Layer 2 domains and connect via Layer 3. VXLAN tunnels are terminated within each data center, and DCI uses Layer 3 routing to forward traffic between VXLAN instances.

The spine configurations of DC1-Spine1 are same as the previous section.
When the fabric is configured with Ethernet Routing Bridging (ERB) and IP prefix advertisement services using Type 5 (T5) EVPN routes at the server leaf level, it often becomes necessary to stretch Layer 3 (L3) contexts alongside Layer 2 (L2). The goal is to extend T5 IP VRFs across data centers without requiring VXLAN tunnels from a leaf in one data center to every leaf in remote sites. Instead, the border leaf at the data center edge serves as the gateway. It terminates existing T5 VXLAN tunnels and establishes new tunnels to the remote data center edge. So, we will have 3 Type-5 tunnels, 2 local to the DC and 1 DCI tunnel.

On Arista's EOS DC1-Leaf1 and DC1-BL is configured with 3333 as the L3VNI and RT 3:3.
Here is the configuration from the Arista's EOS leaf.

A snippet from DC1-Leaf1:

```
DC1-Leaf1#show run
vlan 31
   name VLAN31
!
vrf instance VRF3_IP_VRF
!
interface Ethernet4
   switchport trunk allowed vlan 31
   switchport mode trunk
!
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.1/32
!
interface Loopback1
   description VTEP_vxlan
   ip address 3.1.1.1/32
!
interface Vlan31
   mtu 9200
   vrf VRF3_IP_VRF
   ip address 31.0.0.1/24
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 31 vni 10031
   vxlan vrf VRF3_IP_VRF vni 3333
!
ip routing
ip routing vrf VRF3_IP_VRF
!
ip prefix-list PL-loopback-overlay
   seq 10 permit 151.0.0.0/24 eq 32
   seq 20 permit 3.1.1.0/24 eq 32
!
route-map RM-eBGP permit 10
   match ip address prefix-list PL-loopback-overlay
!
router bgp 65001
   router-id 3.1.1.1
   no bgp default ipv4-unicast
   distance bgp 20 100 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
```

```
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor IPv4-UNDERLAY peer group
   neighbor IPv4-UNDERLAY send-community
   neighbor IPv4-UNDERLAY maximum-routes 9000
   neighbor 151.0.0.11 peer group EVPN-OVERLAY
   neighbor 151.0.0.11 remote-as 65011
   neighbor 151.0.0.11 description Spine-1
   neighbor 151.0.0.12 peer group EVPN-OVERLAY
   neighbor 151.0.0.12 remote-as 65011
   neighbor 151.0.0.12 description Spine-2
   neighbor 161.0.1.20 peer group IPv4-UNDERLAY
   neighbor 161.0.1.20 remote-as 65011
   neighbor 161.0.1.20 description Spine-1
   neighbor 161.0.1.28 peer group IPv4-UNDERLAY
   neighbor 161.0.1.28 remote-as 65011
   neighbor 161.0.1.28 description Spine-2
   redistribute connected route-map RM-eBGP
   !
   vlan 31
      rd 3.1.1.1:31
      route-target both 31:31
      redistribute learned
   !
   address-family evpn
      neighbor EVPN-OVERLAY activate
   !
   address-family ipv4
      no neighbor EVPN-OVERLAY activate
      neighbor IPv4-UNDERLAY activate
   !
   vrf VRF3_IP_VRF
      rd 3.1.1.1:3
      route-target import evpn 3:3
      route-target export evpn 3:3
      router-id 3.1.1.1
      redistribute connected
```

A snippet from DC1-BL:

```
DC1-BL#show run

*snip*

vrf instance VRF3_IP_VRF
!
interface Ethernet1
   mtu 9214
   no switchport
   ip address 161.0.1.5/31
!
interface Ethernet2
   mtu 9214
   no switchport
   ip address 171.0.1.150/31
!
interface Loopback0
   description BGP_Overlay
   ip address 151.0.0.6/32
!
interface Loopback1
   description VTEP_vxlan
   ip address 3.1.1.101/32
!
interface Vxlan1
```

```
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vrf VRF3_IP_VRF vni 3333
!
ip routing
ip routing vrf VRF3_IP_VRF
!
ip prefix-list PL-loopback-overlay
   seq 10 permit 151.0.0.0/24 eq 32
   seq 20 permit 3.1.1.0/24 eq 32
!
route-map RM-eBGP permit 10
   match ip address prefix-list PL-loopback-overlay
!
router bgp 65101
   router-id 151.0.0.6
   no bgp default ipv4-unicast
   distance bgp 20 100 200
   maximum-paths 16 ecmp 16
   neighbor EVPN-OVERLAY peer group
   neighbor EVPN-OVERLAY update-source Loopback0
   neighbor EVPN-OVERLAY bfd
   neighbor EVPN-OVERLAY ebgp-multihop 3
   neighbor EVPN-OVERLAY send-community
   neighbor EVPN-OVERLAY maximum-routes 0
   neighbor EVPN-OVERLAY-dci peer group
   neighbor EVPN-OVERLAY-dci update-source Loopback0
   neighbor EVPN-OVERLAY-dci bfd
   neighbor EVPN-OVERLAY-dci ebgp-multihop 3
   neighbor EVPN-OVERLAY-dci send-community
   neighbor EVPN-OVERLAY-dci maximum-routes 0
   neighbor IPv4-UNDERLAY peer group
   neighbor IPv4-UNDERLAY send-community
   neighbor IPv4-UNDERLAY maximum-routes 9000
   neighbor 3.1.1.102 peer group EVPN-OVERLAY-dci
   neighbor 3.1.1.102 remote-as 65102
   neighbor 3.1.1.102 description DC2-BL
   neighbor 151.0.0.11 peer group EVPN-OVERLAY
   neighbor 151.0.0.11 remote-as 65011
   neighbor 151.0.0.11 description Spine-1
   neighbor 161.0.1.4 peer group IPv4-UNDERLAY
   neighbor 161.0.1.4 remote-as 65011
   neighbor 161.0.1.4 description Spine-1
   neighbor 171.0.1.151 peer group IPv4-UNDERLAY
   neighbor 171.0.1.151 remote-as 65102
   neighbor 171.0.1.151 description DC2-BL
   redistribute connected route-map RM-eBGP
   !
   address-family evpn
      neighbor EVPN-OVERLAY activate
      neighbor EVPN-OVERLAY-dci activate
      neighbor EVPN-OVERLAY-dci domain remote
      neighbor default next-hop-self received-evpn-routes route-type ip-prefix inter-domain
   !
   address-family ipv4
      no neighbor EVPN-OVERLAY activate
      neighbor IPv4-UNDERLAY activate
   !
   vrf VRF3_IP_VRF
      rd 151.0.0.6:3
      route-target import evpn 3:3
      route-target export evpn 3:3
      router-id 151.0.0.6
      redistribute connected
```

Configuring EVPN-VXLAN T5 to EVPN-VXLAN T5 Seamless Stitching on Junos OS/Junos OS Evolved

The only difference in configuring Junos OS DC2-Leaf1 and DC2-BL from the previous section is the addition of the IPVRF . The RT configured is 30:30 and L3VNI is 3333. An irb is also created for VLAN32 to be used by the IPVRF.

On Junos OS, the RT 30:30 is used locally by DC2-Leaf1 and 3:3 is used for DCI by DC2-BL.

A snippet from DC2-Leaf1:

```
root@DC2-Leaf1> show configuration

interfaces {
*snip*

    ge-0/0/4 {
        mtu 9214;
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members 32;
                }
            }
        }
    }
    irb {
        unit 32 {
            family inet {
                address 32.0.0.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 3.1.1.7/32;
            }
        }
        unit 3 {
            family inet {
                address 5.1.1.7/32;
            }
        }
    }
}

policy-options {

*snip*
    policy-statement s3 {
    term 1 {
        from {
            route-filter 32.0.0.0/24 orlonger;
        }
        then accept;
    }
}
routing-instances {
    VRF3_IP_VRF {
        instance-type vrf;
        routing-options {
            multipath;
        }
        protocols {
            evpn {
```

```
                ip-prefix-routes {
                    advertise direct-nexthop;
                    encapsulation vxlan;
                    vni 3333;
                    export s3;
                }
            }
        }
        interface irb.32;
        interface lo0.3;
        route-distinguisher 3.1.1.7:4;
        vrf-target target:30:30;
    }
    VRF3_MAC_VRF_BASED {
        instance-type mac-vrf;
        protocols {
            evpn {
                encapsulation vxlan;
                default-gateway no-gateway-community;
                extended-vni-list all;
            }
        }
        vtep-source-interface lo0.0;
        service-type vlan-based;
        interface ge-0/0/4.0;
        route-distinguisher 3.1.1.7:32;
        vrf-target target:30:30;
        vlans {
            VLAN32 {
             vlan-id 32;
                l3-interface irb.32;
                vxlan {
                    vni 10032;
            }
        }
    }
}
routing-options {
    route-distinguisher-id 3.1.1.7;
    router-id 3.1.1.7;
    autonomous-system 65007;
    forwarding-table {
        export ecmp;
        chained-composite-next-hop {
            ingress {
                evpn;
            }
        }
    }
}
```

A snippet from DC2-BL:

```
root@DC2-BL> show configuration

*snip*
policy-options {
   policy-statement s3 {
    term 1 {
        from {
            route-filter 32.0.0.0/24 orlonger;
        }
        then accept;
    }
    term 2 {
```

```
        from {
            route-filter 31.0.0.0/24 orlonger;
        }
        then accept;
    }
}

routing-instances {
    VRF3_IP_VRF {
        instance-type vrf;
        routing-options {
            multipath;
        }
        protocols {
    evpn {
        interconnect {
            vrf-target target:3:3;
            route-distinguisher 3.1.1.103:300;
        }
        ip-prefix-routes {
            advertise direct-nexthop;
            encapsulation vxlan;
            vni 3333;
            export s3;
        }
    }
}
    interface lo0.3;
        route-distinguisher 3.1.1.102:4;
        vrf-target target:30:30;
    }
}
routing-options {
    route-distinguisher-id 3.1.1.102;
    router-id 3.1.1.102;
    autonomous-system 65102;
    forwarding-table {
        export ecmp;
        chained-composite-next-hop {
            ingress {
                evpn;
            }
        }
    }
}
```

Understanding How Updates are Exchanged Over the DCI

Since in this case a new Type-5 DCI tunnel is used, the router-mac (RMAC), the route distinguisher and the route-target of DC1-Leaf1 will be rewritten at the border leaf which we will see in the output, but the original VNI value is retained. Also, no I-ESI will be used unlike the previous section and IP ECMP will be used instead for load balancing of the traffic. Type-2 MAC/MAC-IP will also be not advertised to remote Ds for site specific VLANs.

Following Control Plane EVPN Updates from DC2 to DC1

Vlan32 is configured on DC2-Leaf1 with the respective irb.32 (32.0.0.0/24) and Host2’s ARP entry is learnt as shown below.

```
root@DC2-Leaf1> show interfaces terse irb.32
Interface               Admin Link Proto    Local                 Remote
irb.32                  up    up   inet     32.0.0.1/24
                                   multiservice
```

```
root@DC2-Leaf1> show arp vpn VRF3_IP_VRF
MAC Address       Address         Name                         Interface               Flags
2c:6b:f5:db:64:f0 32.0.0.16       32.0.0.16                    irb.32 [ge-0/0/4.32]    permanent remote
```

This subnet 32.0.0.0/24 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 TYPE-5 route from DC2-Leaf1.

```
root@DC2-Spine1> show route receive-protocol bgp 3.1.1.7 match-prefix "*32.0.0.0*" extensive

bgp.evpn.0: 77 destinations, 77 routes (77 active, 0 holddown, 0 hidden)
* 5:3.1.1.7:4::0::32.0.0.0::24/248 (1 entry, 1 announced)
     Accepted
     Route Distinguisher: 3.1.1.7:4
     Route Label: 3333
     Overlay gateway address: 0.0.0.0
     Nexthop: 3.1.1.7
     AS path: 65007 I
     Communities: target:30:30 encapsulation:vxlan(0x8) router-mac:2c:6b:f5:8d:63:f0
```

Router mac address 2c:6b:f5:8d:63:f0 belongs to DC2-Leaf1 as shown below. Here the RT is 30:30, RD is 3.1.1.7:4 Next-hop address is of DC2-Leaf1 and the L3-VNI is 3333 which also belongs to DC2-Leaf1.

```
root@DC2-Leaf1> show interfaces irb
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 133, SNMP ifIndex: 505
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514
  Device flags   : Present Running
  Interface Specific flags: Internal: 0x200
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 2c:6b:f5:8d:63:f0, Hardware address: 2c:6b:f5:8d:63:f0
```

The spines will send this to the other leaf switches in DC2, including DC2-BL.
DC2-BL, using Type-5 seamless stitching, will advertise this route to DC1-BL and re-write the RT, Next-Hop and EVPN routermac address as mentioned earlier.
New RT is 3:3
New RD is 3.1.1.103:300
Next-Hop is self (DC2-BL)
New EVPN routermac address is 08:a9:6d:25:08:05

```
root@DC2-BL> show route advertising-protocol bgp 151.0.0.6 table bgp.evpn.0 match-prefix "*32.0.0.0*" extensive
* 5:3.1.1.103:300::0::32.0.0.0::24/248 (1 entry, 1 announced)
 BGP group OVERLAY type External
     Route Distinguisher: 3.1.1.103:300
     Route Label: 3333
     Overlay gateway address: 0.0.0.0
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65102] 65022 65007 I
     Communities: target:3:3 encapsulation:vxlan(0x8) router-mac:08:a9:6d:25:08:05

root@DC2-BL> show interfaces irb | match address:
  Current address: 08:a9:6d:25:08:05, Hardware address: 08:a9:6d:25:08:05
```

In the below output, under "EVPN->IPv4 Imported Prefixes" Type-5 route for 32.0.0.0/24 prefix is successfully imported from DC2-Leaf1 (3.1.1.7) to DC2-BL.
At the same time, under "IPv4->EVPN Exported Prefixes" 32.0.0.0/24 prefix EVPN route status is "DCI Created" which means that DC2-BL has successfully exported Type-5 route to DC1-BL on the DCI link.

```
root@DC2-BL> show evpn ip-prefix-database l3-context VRF3_IP_VRF
L3 context: VRF3_IP_VRF
```

```
IPv4->EVPN Exported Prefixes
Prefix                                      EVPN route status
31.0.0.0/24                                 DC Created
32.0.0.0/24                                 DCI Created
32.0.0.1/32                                 DCI Created
32.0.0.16/32                                DCI Created

EVPN->IPv4 Imported Prefixes
Prefix                                      Etag
31.0.0.0/24                                 0
  Route distinguisher    VNI/Label/SID          Router MAC         Nexthop/Overlay GW/ESI    Route-Status   Reject-Reason
  3.1.1.1:3              3333                   50:00:00:43:40:cf  3.1.1.101                  Accepted       n/a
32.0.0.0/24                                 0
  Route distinguisher    VNI/Label/SID          Router MAC         Nexthop/Overlay GW/ESI    Route-Status   Reject-Reason
  3.1.1.7:4              3333                   2c:6b:f5:8d:63:f0  3.1.1.7                    Accepted       n/a
32.0.0.1/32                                 0
  Route distinguisher    VNI/Label/SID          Router MAC         Nexthop/Overlay GW/ESI    Route-Status   Reject-Reason
  3.1.1.7:4              3333                   2c:6b:f5:8d:63:f0  3.1.1.7                    Accepted       n/a
32.0.0.16/32                                0
  Route distinguisher    VNI/Label/SID          Router MAC         Nexthop/Overlay GW/ESI    Route-Status   Reject-Reason
  3.1.1.7:4              3333                   2c:6b:f5:8d:63:f0  3.1.1.7                    Accepted       n/a
```

DC1-BL receives this route and it will re-advertise the same route and again change the Next-Hop and re-write EVPN routermac address.

Route received from DC2-BL:

```
DC1-BL#show bgp neighbors 3.1.1.102 evpn received-routes route-type ip-prefix 32.0.0.0/24
BGP routing table information for VRF default
Router identifier 151.0.0.6, local AS number 65101
BGP routing table entry for ip-prefix 32.0.0.0/24 remote, Route Distinguisher: 3.1.1.103:300
 Paths: 1 available
  65102 65022 65007
    3.1.1.102 from 3.1.1.102 (3.1.1.102)
      Origin IGP, metric -, localpref -, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:08:a9:6d:25:08:05
      VNI: 3333
```

Type-5 route advertised to DC1-Spine. Here the Routermac 50:00:00:43:40:cf is the local mac address of DC1-BL.

```
DC1-BL#show bgp neighbors 151.0.0.11 evpn advertised-routes route-type ip-prefix 32.0.0.0/24
BGP routing table information for VRF default
Router identifier 151.0.0.6, local AS number 65101
Update wait-install is disabled
BGP routing table entry for ip-prefix 32.0.0.0/24, Route Distinguisher: 3.1.1.103:300
 Paths: 1 available
  65101 65102 65022 65007
    3.1.1.101 from - (0.0.0.0)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:50:00:00:43:40:cf
      VNI: 3333
```

Finally, DC1-Leaf1 receives this Type-5 route from DC1-Spine.

```
DC1-Leaf1#show bgp evpn route-type ip-prefix 32.0.0.0/24
BGP routing table information for VRF default
Router identifier 3.1.1.1, local AS number 65001
BGP routing table entry for ip-prefix 32.0.0.0/24, Route Distinguisher: 3.1.1.103:300
 Paths: 1 available
  65011 65101 65102 65022 65007
    3.1.1.101 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:50:00:00:43:40:cf
      Rx path id: 0x1
      VNI: 3333
```

```
DC1-Leaf1#show ip route vrf VRF3_IP_VRF 32.0.0.0/24
VRF: VRF3_IP_VRF

 B E      32.0.0.0/24 [20/0] via VTEP 3.1.1.101 VNI 3333 router-mac 50:00:00:43:40:cf local-interface Vxlan1
```

Following Control Plane EVPN Updates from DC1 to DC2

Vlan31 is configured on DC1-Leaf1 with the respective irb.31 (31.0.0.0/24) and Host1's ARP entry is learnt as shown below.

```
DC1-Leaf1#show ip interface vlan31 brief
                                                                               Address
Interface      IP Address        Status       Protocol           MTU    Owner
-------------- ----------------- ------------ -------------- ---------- -------
Vlan31         31.0.0.1/24       up           up                9200

DC1-Leaf1#show arp vrf VRF3_IP_VRF
Address          Age (sec)  Hardware Addr   Interface
31.0.0.11          2:47:56  5000.0088.2ff3  Vlan31, Ethernet4
```

DC1-Leaf1 will advertise a Type-5 route to DC1-BL with its own chassis mac address as the Evpn Routermac.
DC1-Leaf1 is configured with RT 3:3 and RD 3.1.1.1:3.

DC1-BL receives this Type-5 route from DC1-Leaf1. The Router in this advertised route is 50:00:00:f6:ad:37 which belongs to DC1-Leaf1.

```
DC1-Leaf1#show version
Arista vEOS-lab
Hardware version:
Serial number: CAB6B568DF5227B7EB798802506AA9D4
Hardware MAC address: 5000.00f6.ad37
System MAC address: 5000.00f6.ad37
*snip*
```

```
DC1-BL#show bgp evpn route-type ip-prefix 31.0.0.0/24
BGP routing table information for VRF default
Router identifier 151.0.0.6, local AS number 65101
BGP routing table entry for ip-prefix 31.0.0.0/24, Route Distinguisher: 3.1.1.1:3
 Paths: 1 available
  65011 65001
    3.1.1.1 from 151.0.0.11 (151.0.0.11)
      Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, best
      Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:50:00:00:f6:ad:37
      Rx path id: 0x1
      VNI: 3333
```

Once DC1-BL receives this route it will re-advertise this route to DC2-BL. Here DC1-BL will change the Next-hop to self and re-write the evpn Routermac to its own MAC.
Note: In Arista's EOS RD and RT will not be rewritten by the border leaf unlike Junos OS. The route will have the MAC RD and RT as originally advertised by DC1-Leaf1 RT 3:3 and RD 3.1.1.1:3.
DC2-BL receives this Type-5 route from DC1-BL over the DCI link. Mac 50:00:00:43:40:cf belongs to DC1-BL

```
root@DC2-BL> show route receive-protocol bgp 151.0.0.6 table VRF3_IP_VRF.evpn.0 match-prefix *31.0.0.0*
extensive

VRF3_IP_VRF.evpn.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
* 5:3.1.1.1:3::0::31.0.0.0::24/248 (1 entry, 1 announced)
     Import Accepted
```

```
     Route Distinguisher: 3.1.1.1:3
     Route Label: 3333
     Overlay gateway address: 0.0.0.0
     Nexthop: 3.1.1.101
     AS path: 65101 65011 65001 I
     Communities: target:3:3 encapsulation:vxlan(0x8) router-mac:50:00:00:43:40:cf
```

```
DC1-BL#show version
Arista vEOS-lab
Hardware version:
Serial number: C753ED86D0BE1430F0A7B41DA6B6BDD3
Hardware MAC address: 5000.0043.40cf
System MAC address: 5000.0043.40cf
*snip*
```

In the below output, under “EVPN->IPv4 Imported Prefixes” Type-5 route for 31.0.0.0/24 prefix is successfully imported from DC1-Leaf1 (3.1.1.101).
At the same time, under “IPv4->EVPN Exported Prefixes” 31.0.0.0/24 prefix EVPN route status is “DC Created” which means that DC2-BL has successfully exported this Type-5 route to DC1-Leaf1 (local VTEP).

```
root@DC2-BL> show evpn ip-prefix-database l3-context VRF3_IP_VRF
L3 context: VRF3_IP_VRF

IPv4->EVPN Exported Prefixes
Prefix                                       EVPN route status
31.0.0.0/24                                  DC Created
32.0.0.0/24                                  DCI Created
32.0.0.1/32                                  DCI Created
32.0.0.16/32                                 DCI Created

EVPN->IPv4 Imported Prefixes
Prefix                                       Etag
31.0.0.0/24                                  0
  Route distinguisher    VNI/Label/SID           Router MAC         Nexthop/Overlay GW/ESI   Route-Status  Reject-Reason
  3.1.1.1:3              3333                    50:00:00:43:40:cf  3.1.1.101                 Accepted      n/a
32.0.0.0/24                                  0
  Route distinguisher    VNI/Label/SID           Router MAC         Nexthop/Overlay GW/ESI   Route-Status  Reject-Reason
  3.1.1.7:4              3333                    2c:6b:f5:8d:63:f0  3.1.1.7                   Accepted      n/a
32.0.0.1/32                                  0
  Route distinguisher    VNI/Label/SID           Router MAC         Nexthop/Overlay GW/ESI   Route-Status  Reject-Reason
  3.1.1.7:4              3333                    2c:6b:f5:8d:63:f0  3.1.1.7                   Accepted      n/a
32.0.0.16/32                                 0
  Route distinguisher    VNI/Label/SID           Router MAC         Nexthop/Overlay GW/ESI   Route-Status  Reject-Reason
  3.1.1.7:4              3333                    2c:6b:f5:8d:63:f0  3.1.1.7                   Accepted      n/a
```

NOTE: The above route export is done because of the below policy option configured on DC2-BL.

```
root@DC2-BL> show configuration | match s3 | display set
set policy-options policy-statement s3 term 1 from route-filter 32.0.0.0/24 orlonger
set policy-options policy-statement s3 term 1 then accept
set policy-options policy-statement s3 term 2 from route-filter 31.0.0.0/24 orlonger
set policy-options policy-statement s3 term 2 then accept

set routing-instances VRF3_IP_VRF protocols evpn ip-prefix-routes export s3
```

DC2-Leaf1 finally receives this Type-5 route of 31.0.0.0/24 from DC2-BL. Here the evpn Routermac will be of DC2-BL.

```
root@DC2-Leaf1> show route table VRF3_IP_VRF.evpn.0 match-prefix *31.0.0.0* extensive

VRF3_IP_VRF.evpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
5:3.1.1.102:4::0::31.0.0.0::24/248 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 3.1.1.102:4
                Next hop type: Indirect, Next hop index: 0
                Address: 0x7caba94
                Next-hop reference count: 3
                Kernel Table Id: 0
                Source: 3.1.1.201
```

```
                Protocol next hop: 3.1.1.102
                Label operation: Push 3333
                Label TTL action: prop-ttl
                Load balance label: Label 3333: None;
                Indirect next hop: 0x2 no-forward INH Session ID: 0
                Indirect next hop: INH non-key opaque: 0x0 INH key opaque: 0x0
                State: <Secondary Active Ext>
                Local AS: 65007 Peer AS: 65022
                Age: 3:07:02    Metric2: 0
                Validation State: unverified
                Task: BGP_65022.3.1.1.201
                Announcement bits (1): 0-VRF3_IP_VRF-EVPN-L3-context
                AS path: 65022 65102 65101 65011 65001 I
                Communities: target:30:30 encapsulation:vxlan(0x8) router-mac:08:a9:6d:25:08:05
                Import Accepted
                Route Label: 3333
                Overlay gateway address: 0.0.0.0
                ESI 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 3.1.1.201
                Primary Routing Table: bgp.evpn.0
                Thread: junos-main
                Indirect next hops: 1
                        Protocol next hop: 3.1.1.102 ResolvState: Resolved
                        Label operation: Push 3333
                        Label TTL action: prop-ttl
                        Load balance label: Label 3333: None;
                        Indirect next hop: 0x2 no-forward INH Session ID: 0
                        Indirect next hop: INH non-key opaque: 0x0 INH key opaque: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 171.0.1.18 via ge-0/0/0.0
                                Session Id: 0
                                3.1.1.102/32 Originating RIB: inet.0
                                  Node path count: 1
                                  Forwarding nexthops: 1
                                        Next hop type: Router
                                        Next hop: 171.0.1.18 via ge-0/0/0.0
                                        Session Id: 140
```

Finally, DC2-Host2 can ping DC1-Host1.

```
root@DC2-Host2> ping 31.0.0.11 count 5 source 32.0.0.16
PING 31.0.0.11 (31.0.0.11): 56 data bytes
64 bytes from 31.0.0.11: icmp_seq=0 ttl=60 time=39.567 ms
64 bytes from 31.0.0.11: icmp_seq=1 ttl=60 time=37.407 ms
64 bytes from 31.0.0.11: icmp_seq=2 ttl=60 time=38.239 ms
64 bytes from 31.0.0.11: icmp_seq=3 ttl=60 time=38.908 ms
64 bytes from 31.0.0.11: icmp_seq=4 ttl=60 time=38.456 ms

--- 31.0.0.11 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 37.407/38.515/39.567/0.717 ms
```