Hewlett Packard Enterprise

HPE aruba networking

AOS-CX 10.13 Update: EVPN-VXLAN shared border VTEP Multi-Fabric VLAN extension, L3 multicast routing

Arnaud LE GALL, CSE Aruba

Agenda

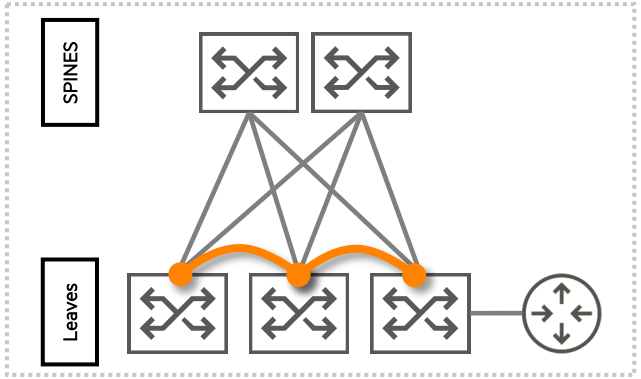
1	Overview
2	Use Cases
3	Details
4	Configuration
5	Best Practices
6	Troubleshooting
7	Demo
8	Scalability and Performances

Overview

Overview
Reminder

Terminology

Aruba uses the following terminology, based on the scope of the Fabrics :

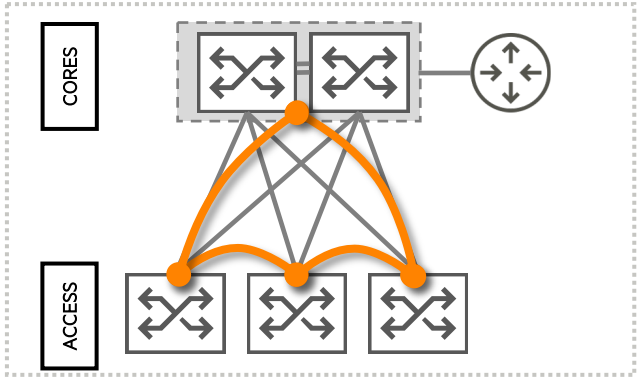


DATACENTER : LEAVES AND SPINES

As a Best Practice, Each Spine is an independent device, only connected to VTEPs

There's no direct connection between Spines.

Spines are pure Layer 3 devices, without any Tunnel termination, and so without any endpoints directly connected to them.



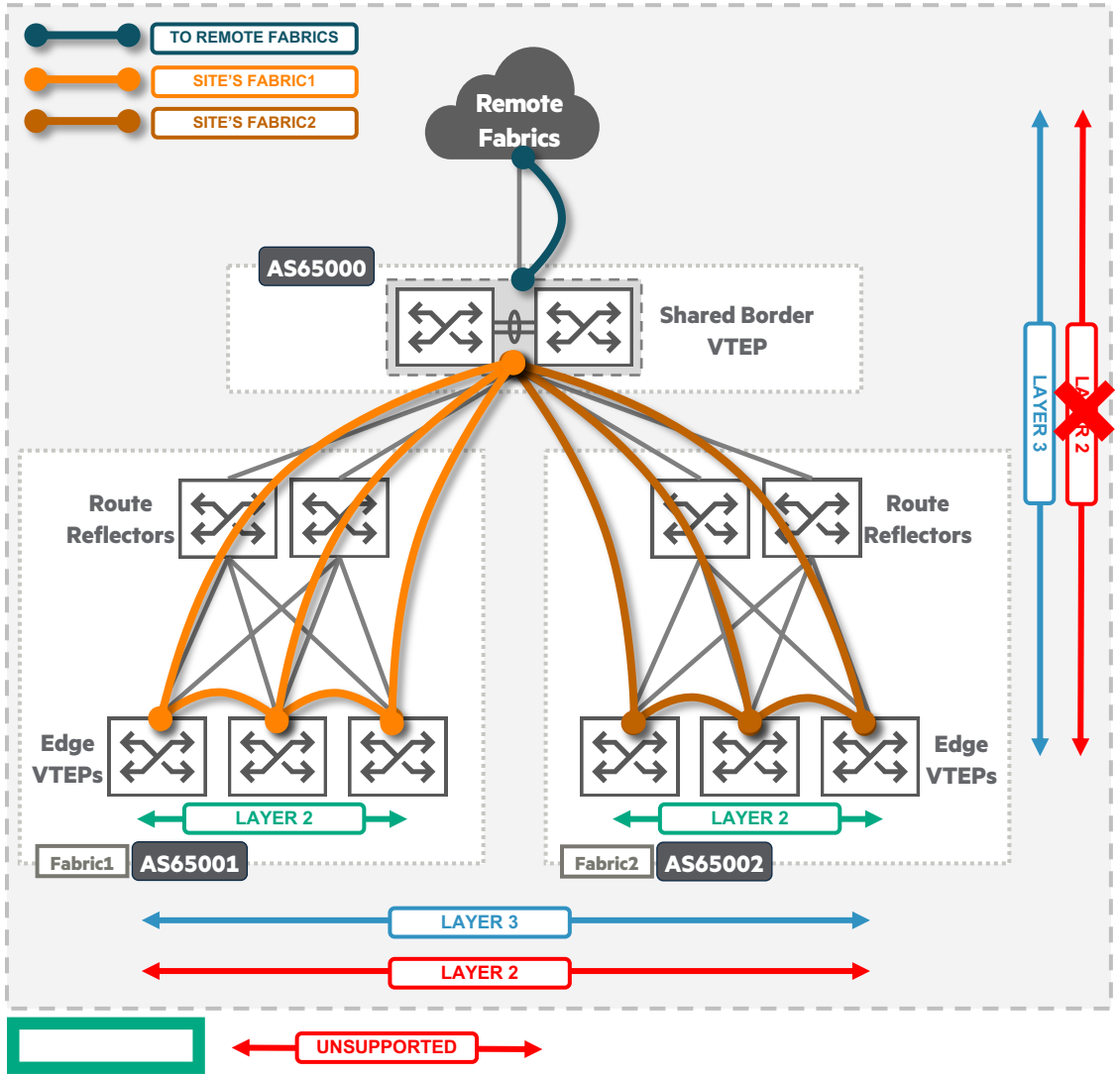
CAMPUS : CORES AND ACCESS

As a Best Practice, Cores will be connected to each other.

This allows to create a logical VSX VTEP at the core to connect endpoints or FW to become a border. Those endpoints will be redistributed in the Fabric.

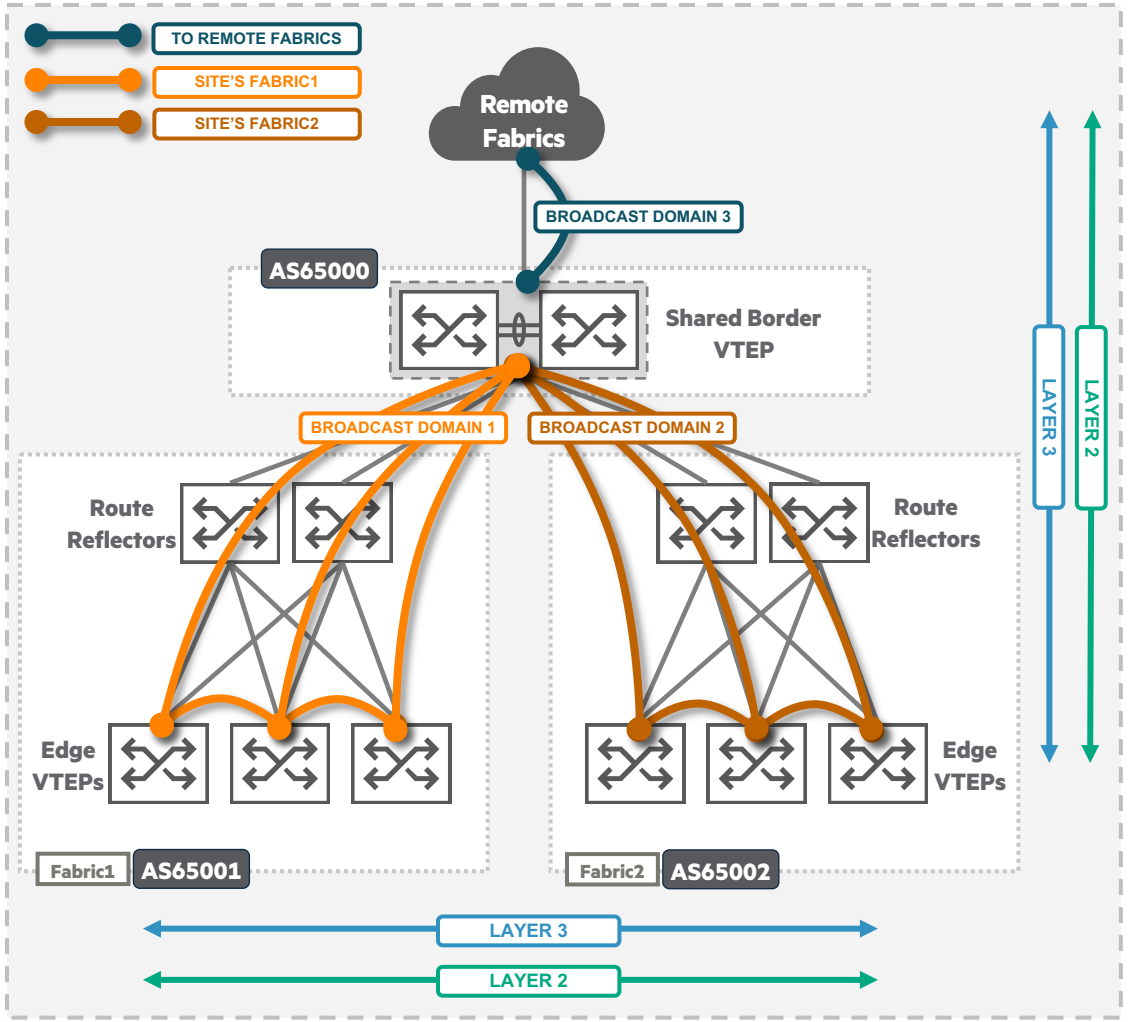
Unicast – VLAN Extension between EVPN-VXLAN Fabrics

Overview
Reminder



- Shared Border topology allows multiple EVPN-VXLAN Fabrics to share a single Border VTEP (Standalone or VSX cluster)
- Since AOS-CX 10.10, only L3VNI can be used between multiple VXLAN Fabrics sharing the same border. L2VNI stretch was restricted within the fabric, between internal VTEPs.
- *Please refer to 10.10 ToI "Aruba AOS-CX 10.10 - Campus Shared Border VTEP" for more details.*

Overview



Since AOS-CX 10.10 and before AOS-CX 10.13

VLAN extension was restricted within the fabric, between internal VTEPs.

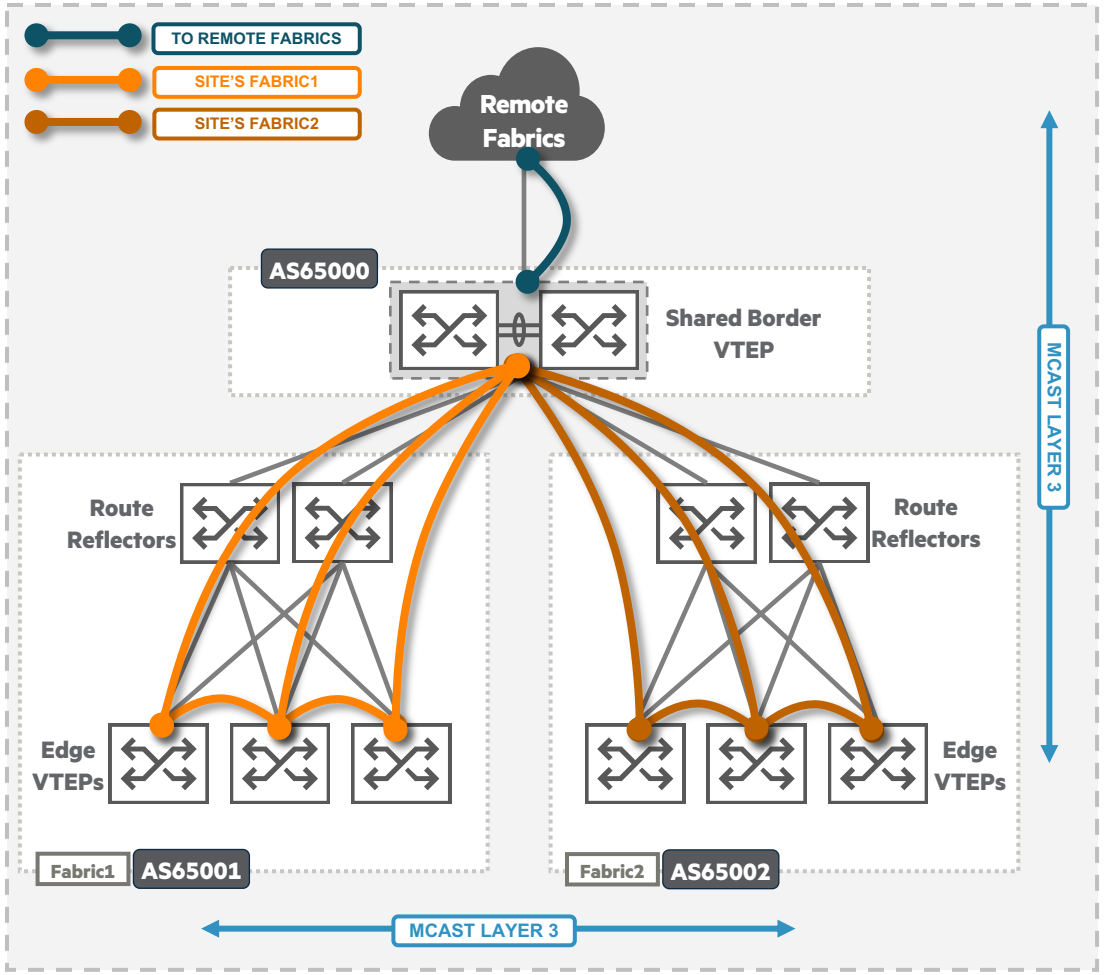
- BUM Traffic is not able to be forwarded between eBGP Fabrics, as tunnels provisioned via all eBGP neighbors share the same broadcast group.
- Split Horizon is only disabled between iBGP and eBGP Tunnels ("dyn-inter-vxlan-bridging ibgp-ebgp")

Since AOS-CX 10.13

- Each Fabric can be configured with a separate broadcast domain, allowing BUM to be forwarded across different eBGP Fabrics, and legacy traffic to be forwarded between eBGP Tunnels.
- A new Route Map attribute called "Local Broadcast Groups" allows to define those Broadcast Domains and to map them with VXLAN Tunnels.
- **New associated feature :**
  - **Broadcast Group ID setting through BGP route-map command**

Multicast – Layer 3

Overview



Before 10.13

Multicast Traffic (Layer 2 or Layer 3) between Fabrics attached to a Shared Border was not supported.

Since AOS-CX 10.13

Shared Border topology now supports **Layer 3 Multicast** between different Fabrics attached to the same Border.

New associated features :

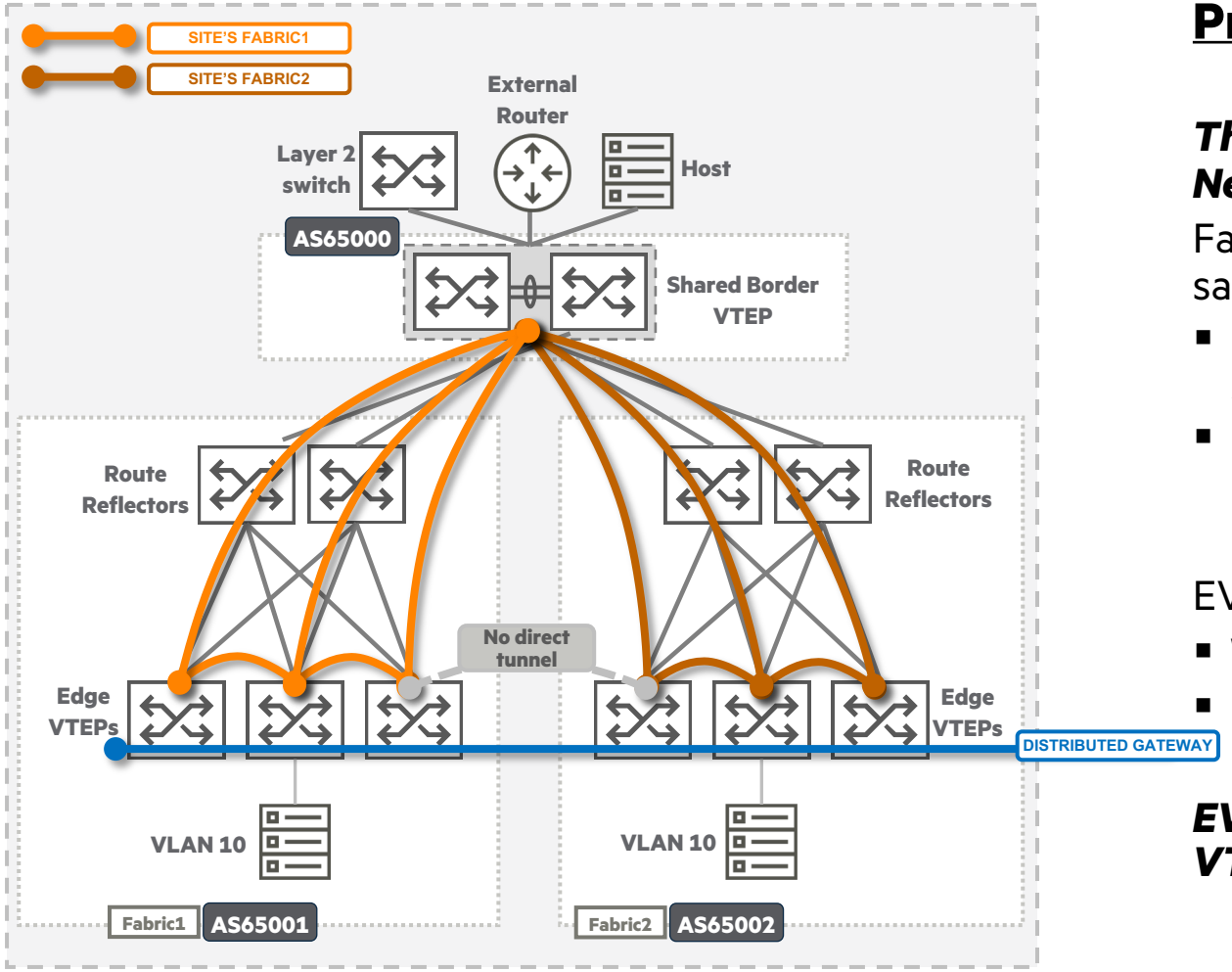
- **Multicast Inter-Fabric capability can be configured on the Shared Border**
- **Multicast Fabric Identifier attached to each Fabric**

Use Cases

Unicast - VLAN Extension between EVPN-VXLAN Fabrics

Use Cases

Validated use cases – Unicast - Multi-Fabric / Single Site – Distributed Gateways



Primary use case

This use case is supported on both Campus and DC Networking.

Fabric 1 and Fabric 2 share a "Border VTEP", as part of the same site :

- Layer 2 Subnets must be stretched over Fabric 1 and Fabric 2.
- Distributed Gateway (Symmetric IRB) handles routing across Fabrics

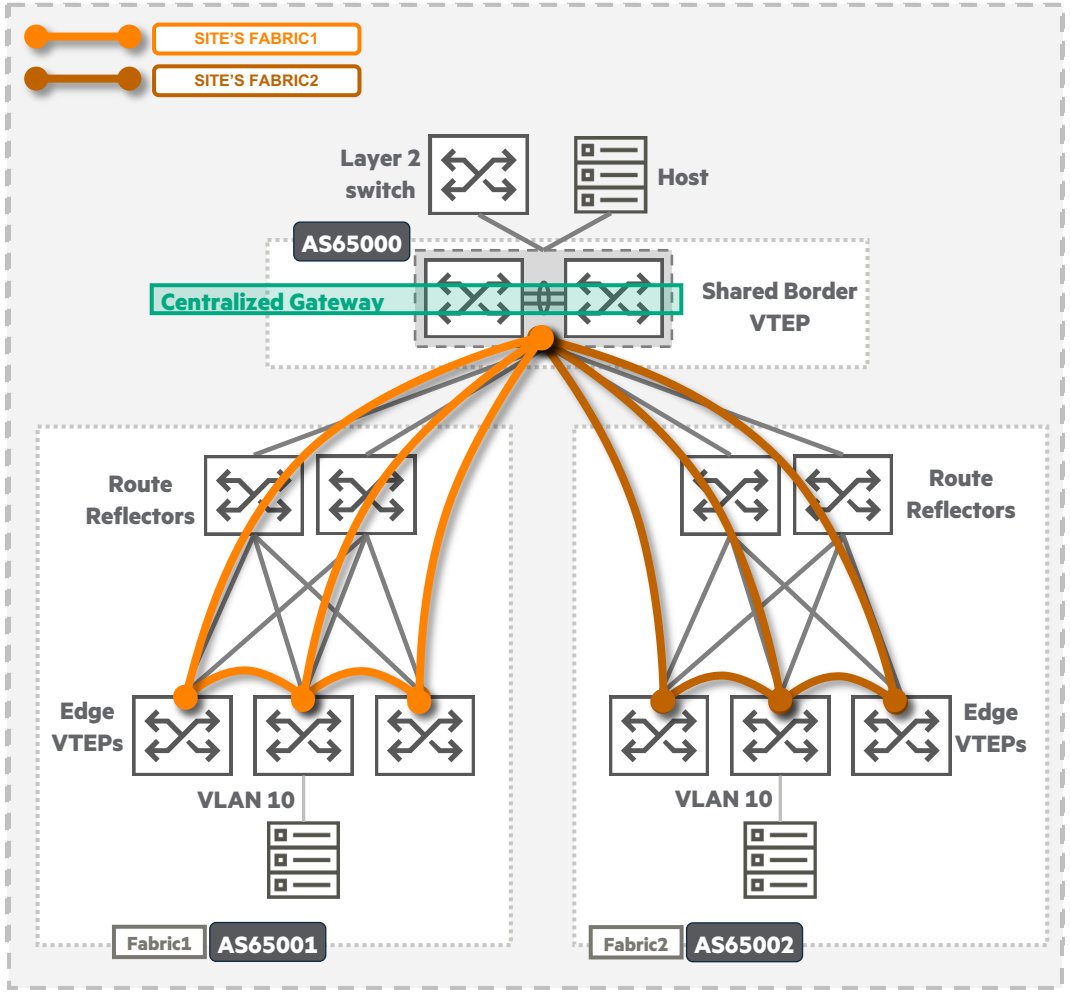
EVPN-VXLAN tunnels must be provisioned:

- Within each Fabric
- Between Shared Border and VTEPs of each Fabrics

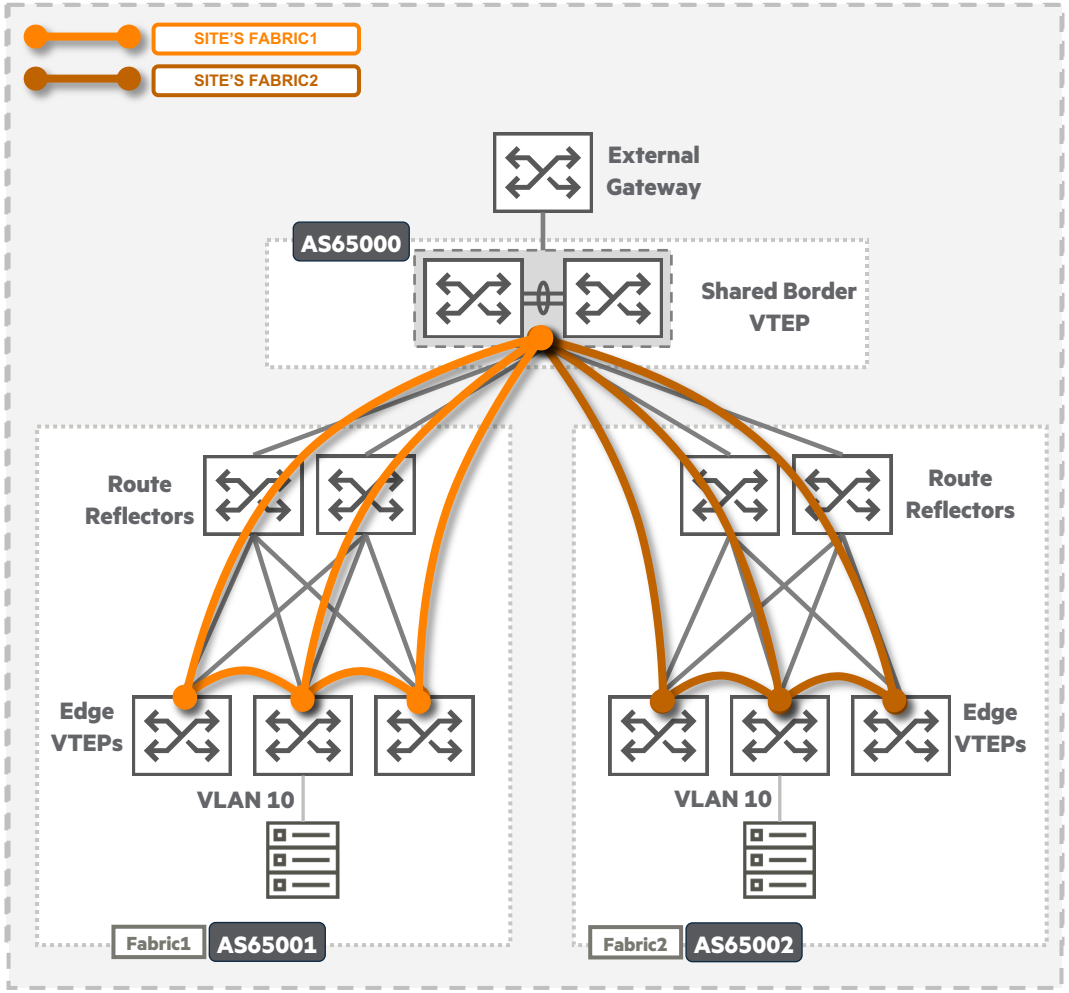
EVPN-VXLAN tunnels must not be provisioned between VTEPs of different Fabrics

Use Cases

Validated use cases – Unicast - Multi-Fabric / Single Site



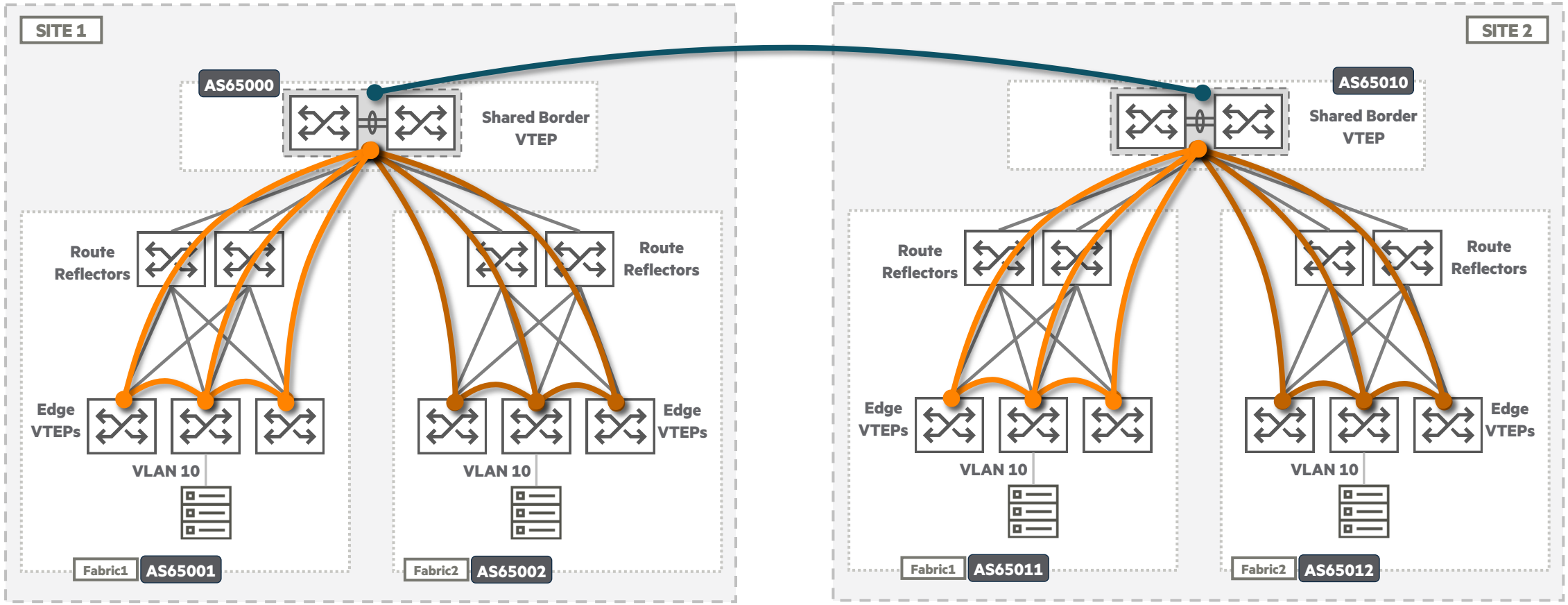
Use Case : Centralized Gateway located on Shared Border.



Use Case : Pure EVPN L2 Fabric with external Gateway (No routing in the Fabrics)

Use Cases

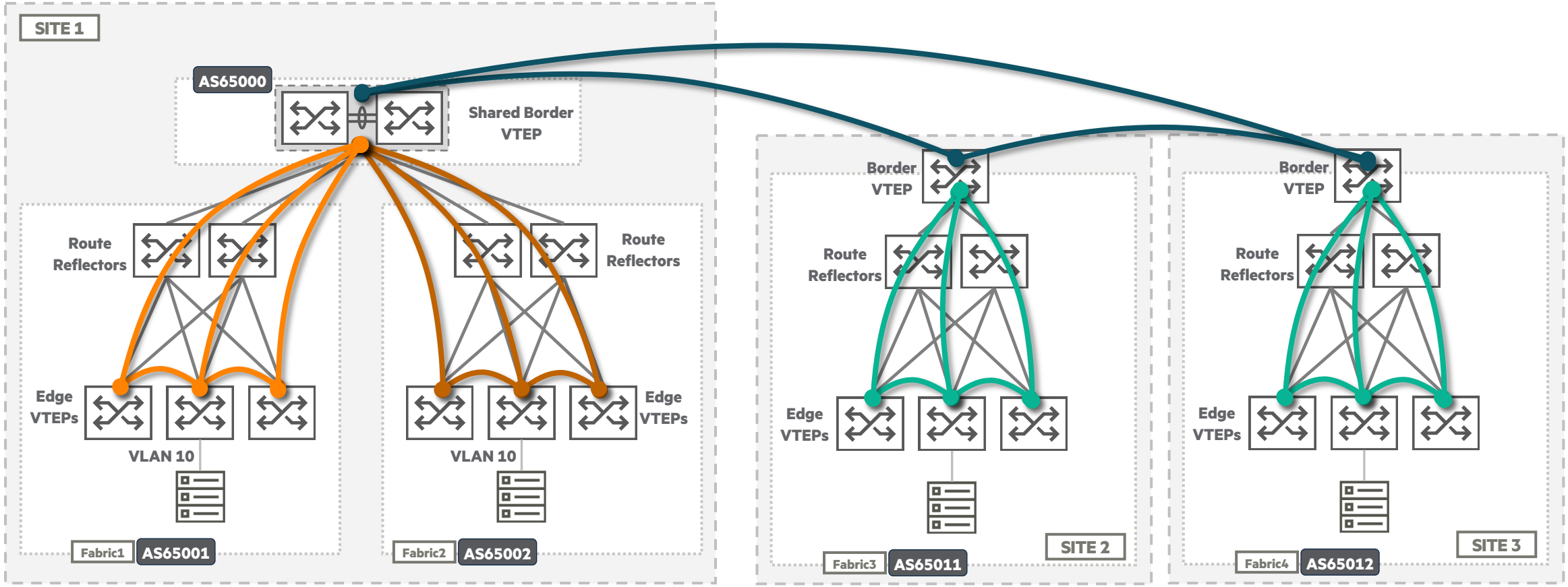
Validated use cases – Unicast - Multi-Fabric / Multiple Sites



Use Case : Multi-Fabric with Shared Border on both sites. L2 Subnets stretched across the local and remote Fabrics.

Use Cases

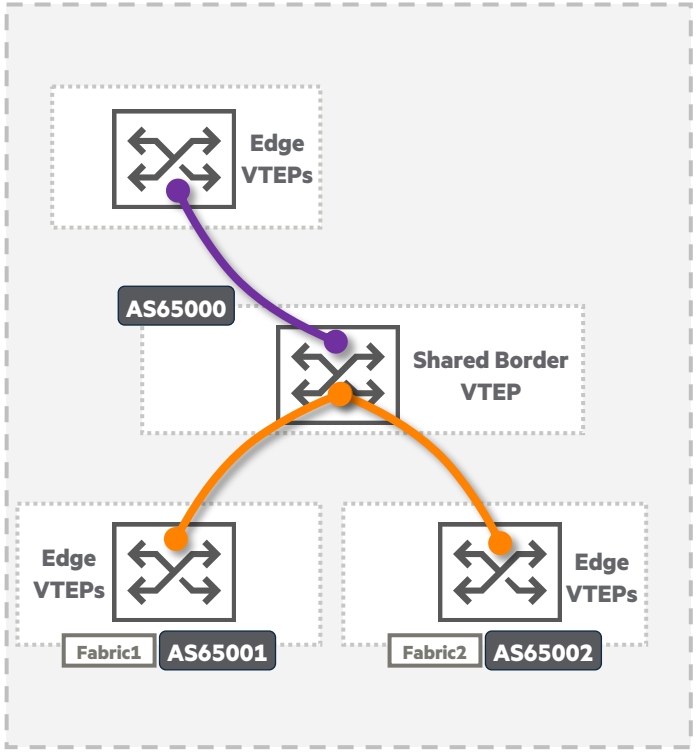
Validated use cases – Unicast - Multi-Fabric / Multiple Sites



Use Case : Multi-Fabric with Shared Border on at least on site. L2 Subnets stretched across the local and remote Fabrics.

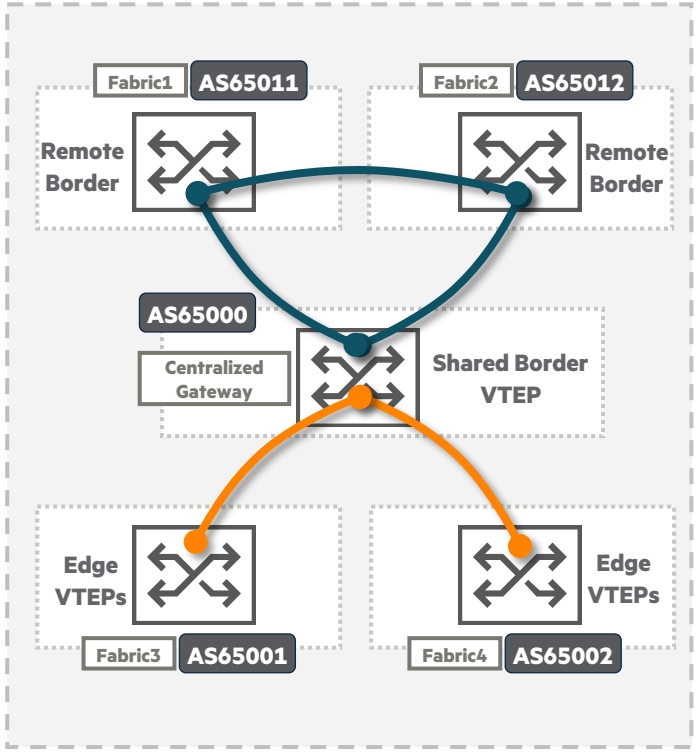
Use Cases

Unicast - Unsupported topologies



DOES NOT WORK

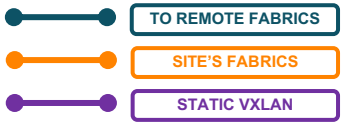
Use Case : Shared Border used as a Stub VTEP, with a Static VXLAN Tunnel to a remote VTEP.



NOT SUPPORTED

Use Case : Multi-Fabric, with Centralized Gateway for Remote Fabrics.

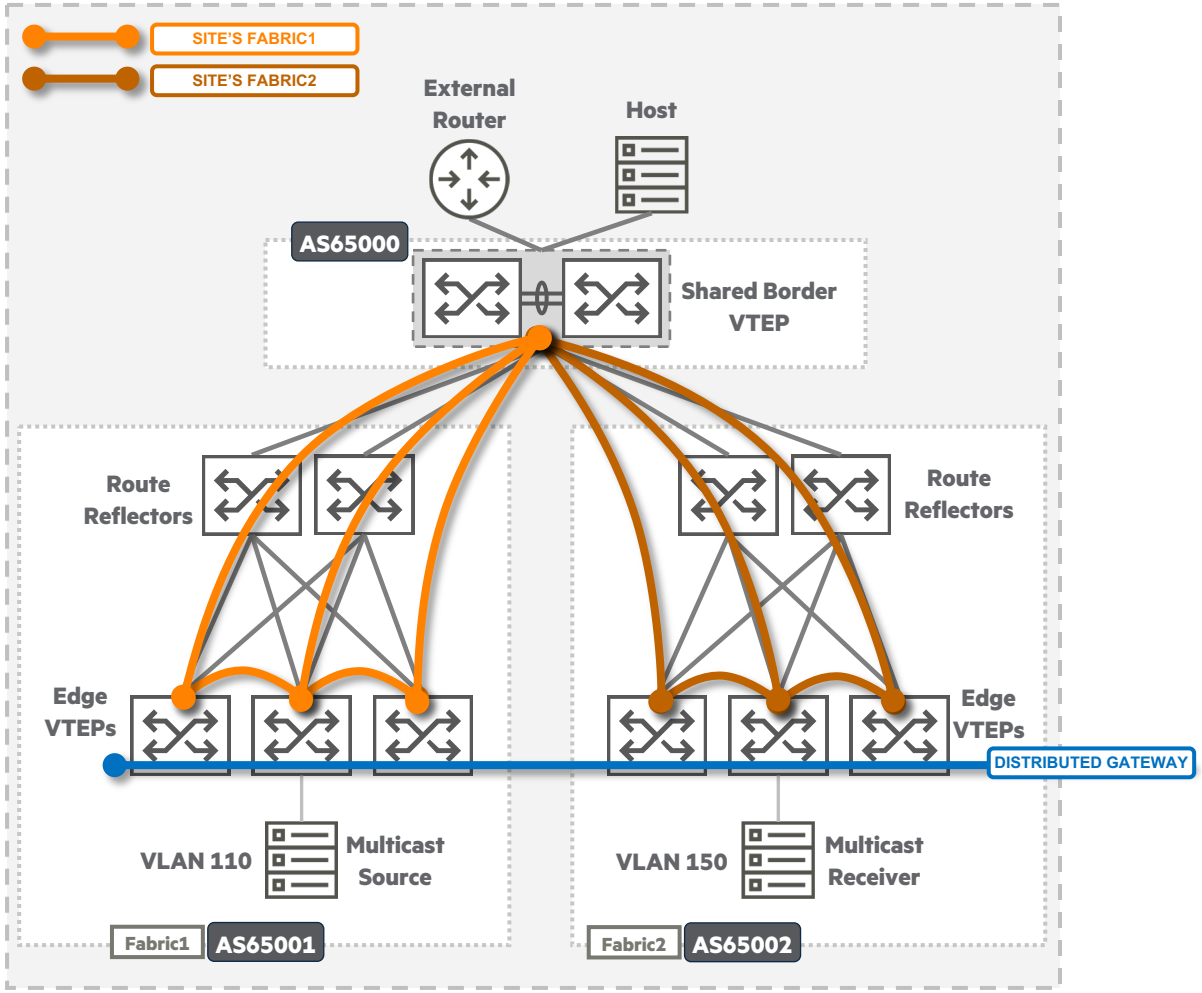
Even if it may work technically, this centralized architecture is not recommended as not optimized. Hence, not tested and unsupported.



Multicast – Layer 3

Use Cases

Validated use cases – Multicast - Multi-Fabric / Single Site



Primary use case

Fabric 1 and Fabric 2 share a "Border VTEP", as part of the same site :

- Source and Receivers are in different Fabrics
- Source and Receivers are in different VLAN

Layer 3 Multicast traffic must be forwarded between Source and Receivers

L3 Multicast Traffic can be from/to another Fabric attached to the Shared Border, or any other remote Fabric.

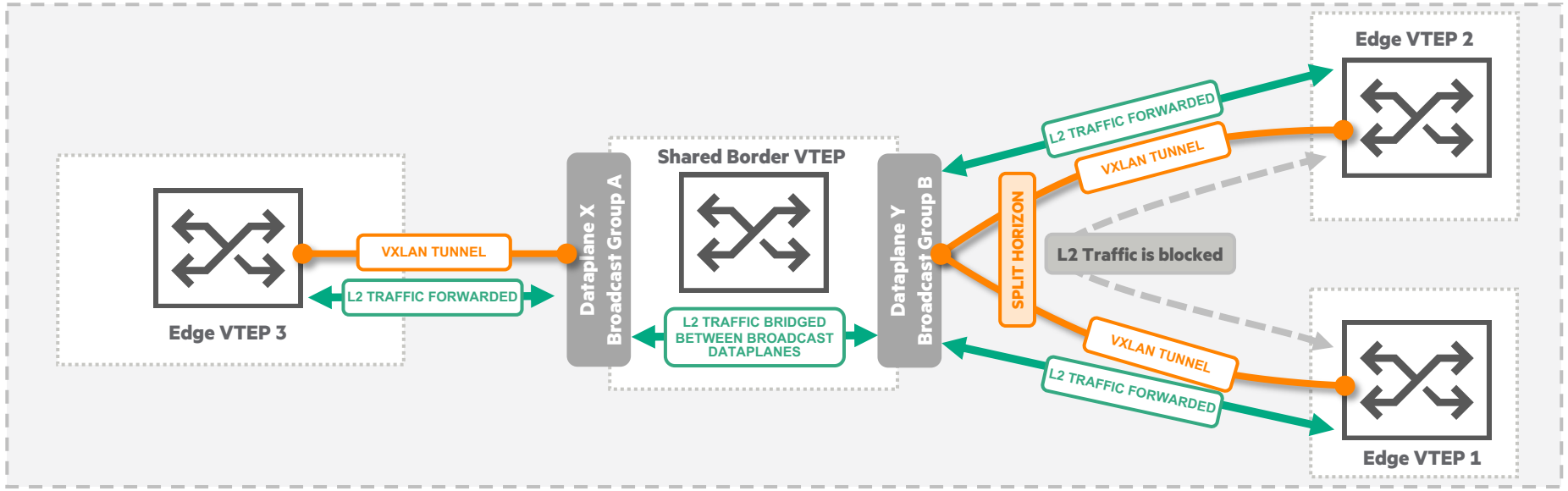
Details

Broadcast Dataplanes

Details

Broadcast Group – Broadcast Dataplane

- AOS-CX 10.13 introduces the capability for each VXLAN Tunnel to be attached to a specific Broadcast Dataplane.
  - A Dataplane ID is a value used inside the ASIC, to categorize a packet to forward it (or not) between tunnels.
  - Once a VTEP receives a Type-3 IMET EVPN route from a BGP peer, and establishes the corresponding VXLAN tunnel, a Broadcast Dataplane ID is associated to the VXLAN tunnel.
  - This Broadcast Dataplane ID is set to 1 by default.
- With Split Horizon Rule, a Border cannot forward traffic between VTEPs Tunnels sharing the same Dataplane ID.
- A shared border can forward L2 traffic between VTEPs whose tunnels are mapped to different Broadcast Group IDs



Details

Broadcast Group – Broadcast Dataplane

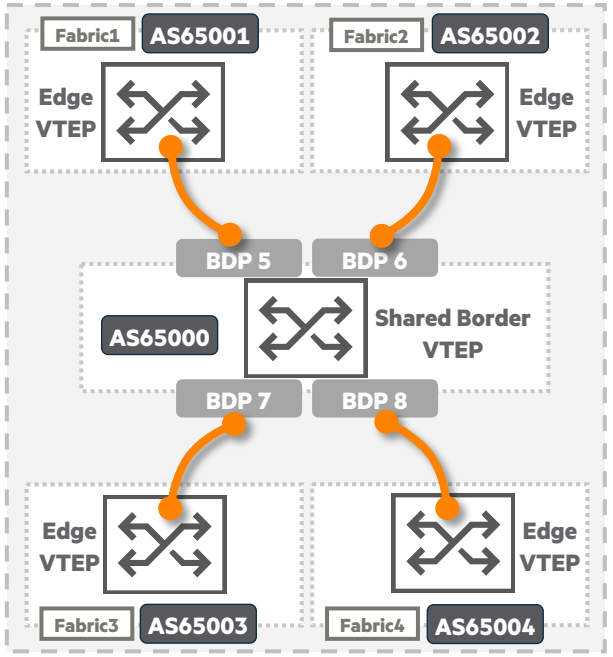
- To relax Split Horizon between VTEPs/Fabrics, they must be mapped to different Broadcast Dataplanes.
  - **With "dyn-vxlan-tunnel-bridging-mode" enabled** :
    - iBGP-based tunnels will be placed in Broadcast Dataplane #1
    - eBGP-based tunnels will be placed in Broadcast Dataplane #4
    - The Dataplane ID cannot be modified.
  - **With "dyn-vxlan-tunnel-bridging-mode" disabled (default)** :
    - Default : All tunnels are placed in Broadcast Dataplane 1
    - This Dataplane ID can be modified using the Broadcast Group ID.
      - **On 8325** : Dataplane ID = Broadcast-Group ID + Offset (+3)
        - Example : if configured Broadcast-Group ID =2, so Dataplane ID = 5
      - **On 8360** : Dataplane ID = Broadcast-Group ID + Offset (+127)
        - Example : if configured Broadcast-Group ID =2, so Dataplane ID = 129

- Broadcast Dataplanes allow to bridge traffic between every type of tunnel :
  - eBGP to eBGP / iBGP to iBGP / eBGP to iBGP Tunnels
  - This new capability could deprecate the "dyn-vxlan-tunnel-bridging-mode" in the future, which is just related to iBGP-eBGP tunnels.
- The Broadcast Group ID configuration only applies to Dynamic Tunnels – Static VXLAN are not concerned.

In AOS-CX 10.13, manually configured Broadcast-Groups are only supported on Shared Border.

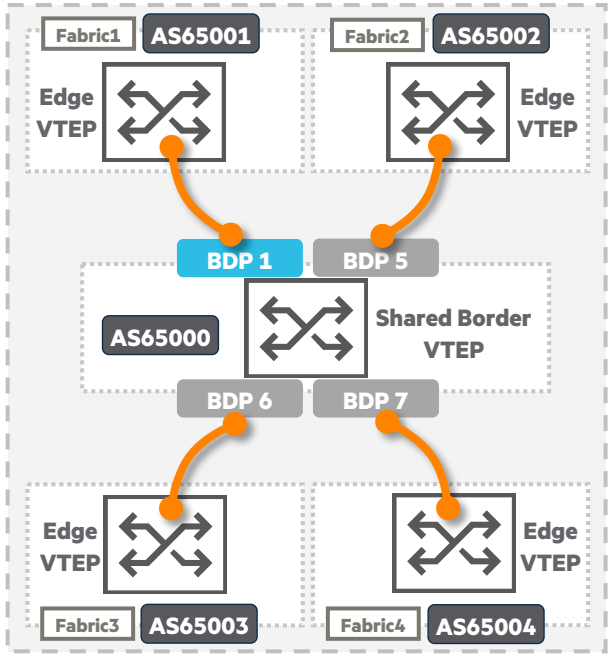
Details

Broadcast Dataplane – Examples with Aruba 8325 as Shared Border



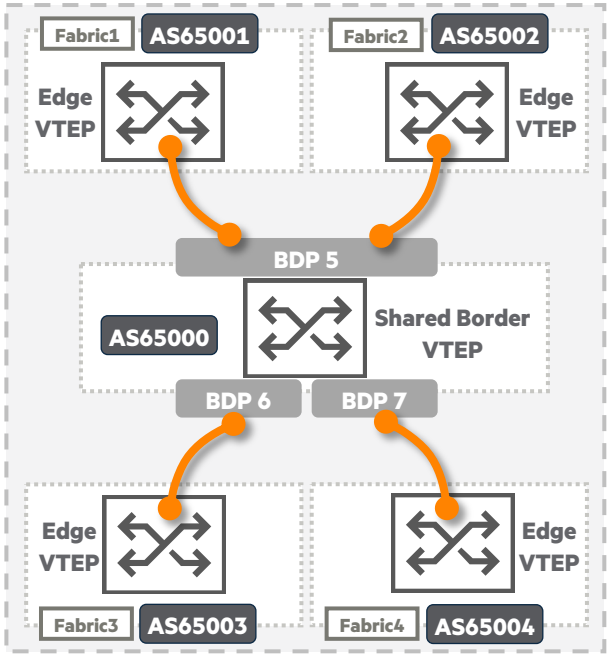
WILL WORK

L2 Traffic can be stretched across all Fabrics, because their attached to different Broadcast Dataplanes.



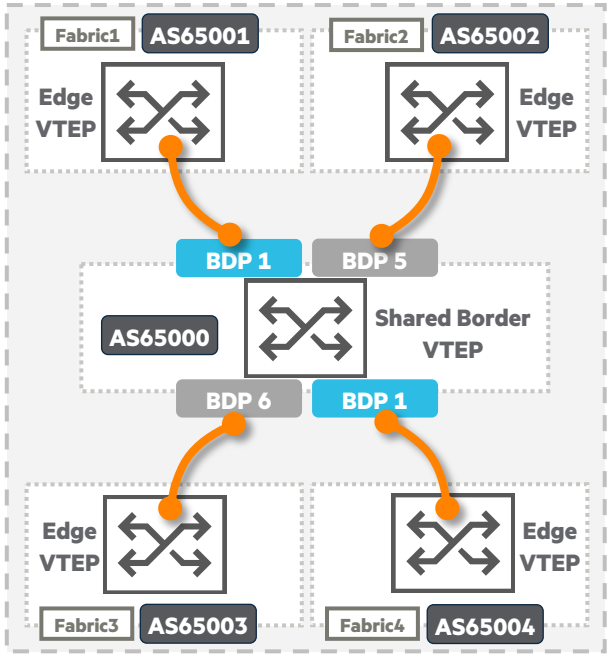
WILL WORK

L2 Traffic can be stretched across all Fabrics, because they're attached to different Broadcast Dataplanes, Fabric 1 being attached to Broadcast Dataplane 1



WILL PARTIALLY WORK

Fabric 1 and Fabric 2 won't be able to get subnet stretched, as they're part of the same Broadcast Dataplane, while L2 Traffic between Fabric 3 and Fabric 4 will be forwarded. Fabric 1 and 2 to Fabric 3 and 4 will work as well.



WILL PARTIALLY WORK

Fabric 1 and Fabric 4 won't be able to get subnet stretched, as they're part of the same Broadcast Dataplane (ID = 1) , while L2 Traffic between Fabric 2 and Fabric 3 will be forwarded. Fabric 1 and 4 to Fabric 2 and 3 will work as well.

BDP X	Default Broadcast Dataplane ID (ID=1)
BDP X	Broadcast Dataplane with configured Broadcast Group

Reminder : *Dataplane ID = Broadcast Group ID + Offset*

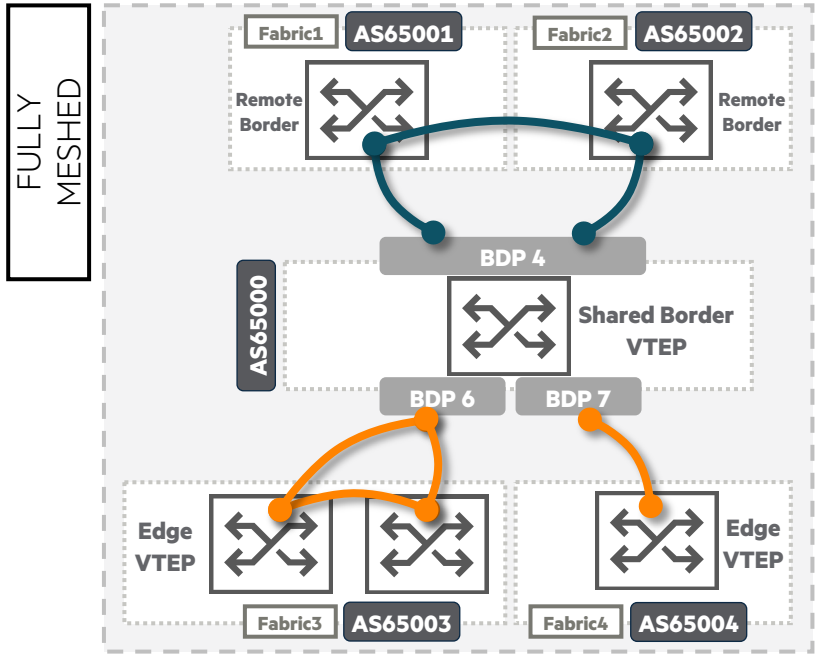
No Broadcast Group configured = Dataplane ID set to 1

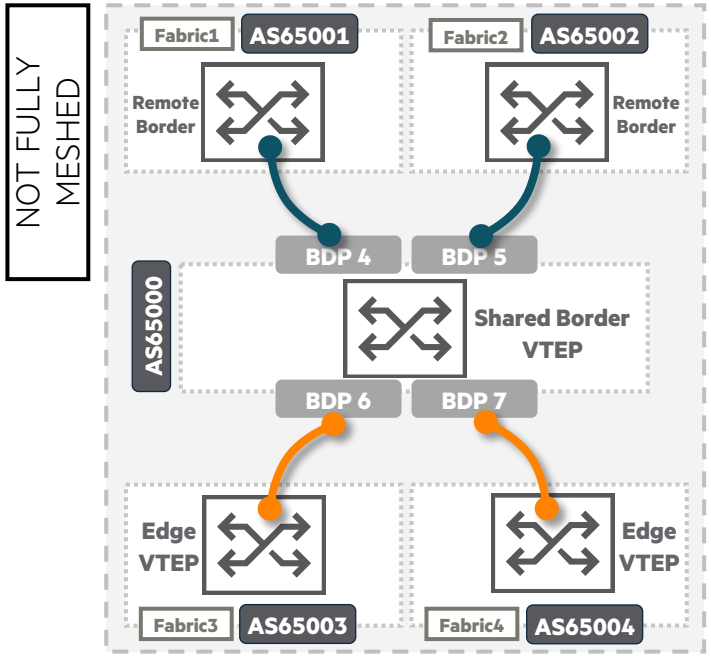
Details

Remote Fabrics

When a shared border peers with remote dedicated borders:

- All the tunnels to the remote border VTEPs are part of the same Broadcast Group, as full mesh tunnels are provisioned by design between all remote borders.
- All VXLAN tunnels towards the internal VTEPs of separate fabrics should be part of different Broadcast Groups.
  - If not, BUM Traffic will not be forwarded between different Fabrics, due to VXLAN split horizon rule
- Reminder: All VXLAN tunnels towards the internal VTEPs of the same fabric are part of the same Broadcast Dataplane.





Platform support

Platform Support – As a Shared Border VTEP

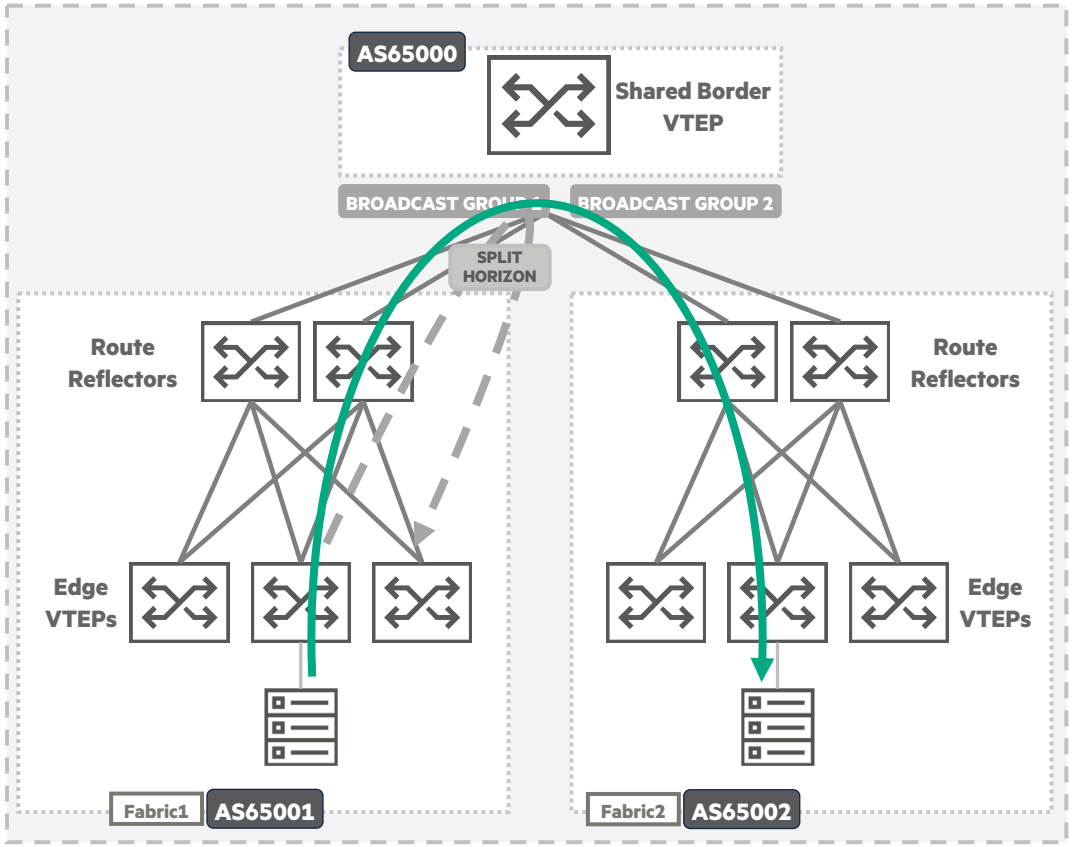
PLATFORMS	6000/6100	6200	6300/6400	8100	8360	8325	10000	9300	8400	OVA
L2 VLAN Extension Supported	No	No	No	No	Yes	Yes	No (PoC-Only*)	No	No	No
L3 MCAST Routing Supported	No	No	No	No	Yes	Yes	No (PoC-Only*)	No	No	No
Max Broadcast Groups	N/A	N/A	N/A	N/A	128	4	4	N/A	N/A	N/A

* Aruba 10000 can be used today as a Border VTEP, with the following points to note:

- L2 Extension can be implemented, and is supported, on Aruba 10000 used as Border.
- Hosts can be directly connected to the Aruba 10000, and Filtering capabilities brought by the AMD Pensando chipset can be used for those endpoints.
- Filtering capabilities cannot be used between 2 VXLAN Tunnels, meaning for traffic coming from an Edge VTEP in Fabric 1 to an Edge VTEP in Fabric 2 or to a remote Fabric (and vice-versa).
- **The newly introduced command "uplink-to-uplink", located in the DSM configuration context, must be used to avoid to forward the traffic received from and sent to ports with persona "Uplink" to the AMD Pensando chipsets.**

Unicast - VLAN Extension between EVPN-VXLAN Fabrics

Details
Broadcast Groups



Statement

All BUM Traffic originating from a peer/VTEP in a Fabric, on a given Broadcast Domain, will never be forwarded to another peer/VTEP on that same Fabric / Broadcast Domain (Split Horizon)

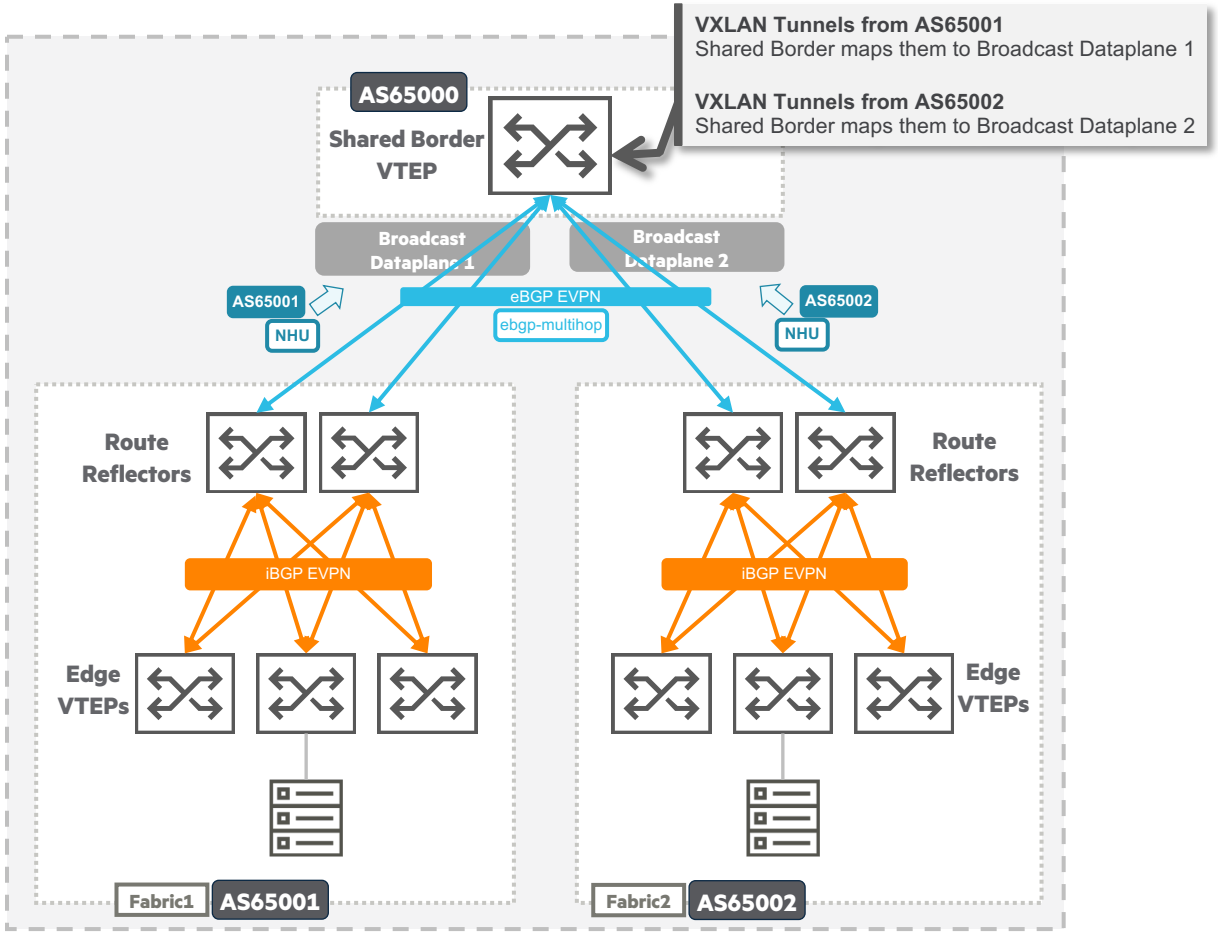
- This implies that a Shared Border will not be able to forward BUM Traffic between 2 attached Fabrics.
- "dyn-vxlan-tunnel-bridging-mode ibgp-ebgp" command cannot be used in such a case (as both Fabric1 and Fabric2 EVPN routes come as eBGP routes)

Solution

Associate each Fabric with a **Broadcast Group** at the Shared Border level and relax Split Horizon between those Groups ("*Flexible Selective Split Horizon*")

- Allows BUM Traffic forwarding across different Broadcast Groups
- A Route Map must be created, allowing to set the same broadcast group for all tunnels towards VTEPs in a specific fabric.

Details



Configuration adjustment is simple and minimum :

- Configuration applies only on the Shared Border
- A new inbound route-map sequence is applied on EVPN Peers of each Fabrics (Route-Reflectors)

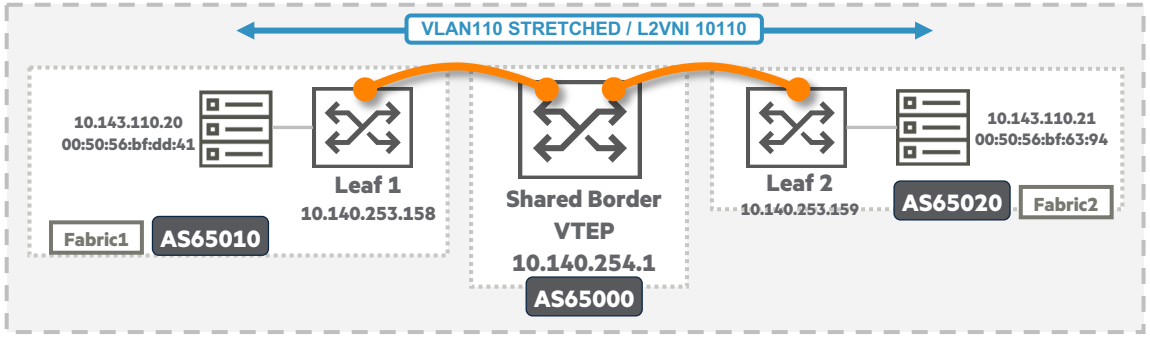
Once done, all IMET Type-3 EVPN Routes coming from those peers will be mapped to their related configured Broadcast Group and hence the provisioned VXLAN tunnels:

- Traffic coming from a VXLAN tunnel in Fabric 1 will not be forwarded to another VXLAN tunnel on the same Dataplane.
- Traffic coming from a VXLAN tunnel in Fabric 1 will be forwarded to another VXLAN tunnel in a different Dataplane (i.e. to Fabric2 VXLAN tunnel)

For educative purpose

Use of a route-map to apply a Broadcast Group ID, instead of a peering attribute, is more flexible, as it allows to apply filters to incoming routes.

Details



BORDER

```
Border(config)# show mac-address-table
MAC age-time            : 300 seconds
Number of MAC addresses : 11

MAC Address          VLAN     Type                      Port
-----------------------------------------------------------------
00:50:56:bf:63:94    110      evpn                      vxlan1(10.147.253.159)
00:50:56:bf:dd:41    110      evpn                      vxlan1(10.147.253.158)
< ------ snipped ------ >

Border(config)# show bgp l2vpn evpn vni 10110
< ------ snipped ------ >

     Network                                          Nexthop              Metric     LocPrf    Weight   Path
-------------------------------------------------------------------------------------------------------------
Route Distinguisher: 10.147.253.158:110      (L2VNI 10110)
*>e [2]:[0]:[0]:[00:00:00:00:01:10]:[10.143.110.254]      10.147.253.158       0          100        0       65010 ?
*>e [2]:[0]:[0]:[00:50:56:bf:dd:41]:[10.143.110.20]       10.147.253.158       0          100        0       65010 ?
*>e [2]:[0]:[0]:[00:50:56:bf:dd:41]:[]                    10.147.253.158       0          100        0       65010 ?
*>e [3]:[0]:[10.147.253.158]                              10.147.253.158       0          100        0       65010 ?
Route Distinguisher: 10.147.253.159:110      (L2VNI 10110)
*>e [2]:[0]:[0]:[00:00:00:00:01:10]:[10.143.110.254]      10.147.253.159       0          100        0       65020 ?
*>e [2]:[0]:[0]:[00:50:56:bf:63:94]:[10.143.110.21]       10.147.253.159       0          100        0       65020 ?
*>e [2]:[0]:[0]:[00:50:56:bf:63:94]:[]                    10.147.253.159       0          100        0       65020 ?
*>e [3]:[0]:[10.147.253.159]                              10.147.253.159       0          100        0       65020 ?
< ------ snipped ------ >
```

LEAF 1

```
Leaf-1(config)# show mac-address-table
MAC age-time            : 300 seconds
Number of MAC addresses : 10

MAC Address          VLAN     Type                      Port
-----------------------------------------------------------------
00:50:56:bf:63:94    110      evpn                      vxlan1(10.147.254.1)
< ------ snipped ------ >

Leaf-1(config)# show bgp l2vpn evpn vni 10110
< ------ snipped ------ >

     Network                                          Nexthop              Metric     LocPrf    Weight   Path
-------------------------------------------------------------------------------------------------------------
Route Distinguisher: 10.147.253.159:110      (L2VNI 10110)
*>i [2]:[0]:[0]:[00:50:56:bf:63:94]:[10.143.110.21]       10.147.254.1         0          100        0       65000 65020 ?
*>i [2]:[0]:[0]:[00:50:56:bf:63:94]:[]                    10.147.254.1         0          100        0       65000 65020 ?
*>i [3]:[0]:[10.147.253.159]                              10.147.254.1         0          100        0       65000 65020 ?
< ------ snipped ------ >
```

LEAF 2

```
Leaf-2(config)# show mac-address-table
MAC age-time            : 300 seconds
Number of MAC addresses : 10

MAC Address          VLAN     Type                      Port
-----------------------------------------------------------------
00:50:56:bf:dd:41    110      evpn                      vxlan1(10.147.254.1)
< ------ snipped ------ >

Leaf-2(config)# show bgp l2vpn evpn vni 10110
< ------ snipped ------ >

     Network                                          Nexthop              Metric     LocPrf    Weight   Path
-------------------------------------------------------------------------------------------------------------
Route Distinguisher: 10.147.253.159:110      (L2VNI 10110)
*>i [2]:[0]:[0]:[00:50:56:bf:dd:41]:[10.143.110.20]       10.147.254.1         0          100        0       65000 65010 ?
*>i [2]:[0]:[0]:[00:50:56:bf:dd:41]:[]                    10.147.254.1         0          100        0       65000 65010 ?
*>i [3]:[0]:[10.147.253.158]                              10.147.254.1         0          100        0       65000 65010 ?
< ------ snipped ------ >
```

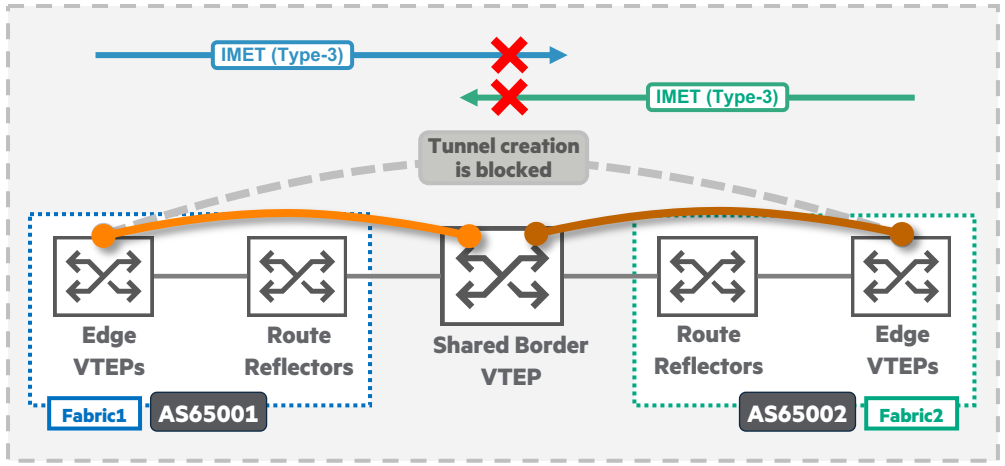
Details

- In AOS-CX 10.13:
  - Both IPv4 and IPv6 in the VXLAN Overlay are supported
- Both VLAN-Based and VLAN-Aware bundles are supported

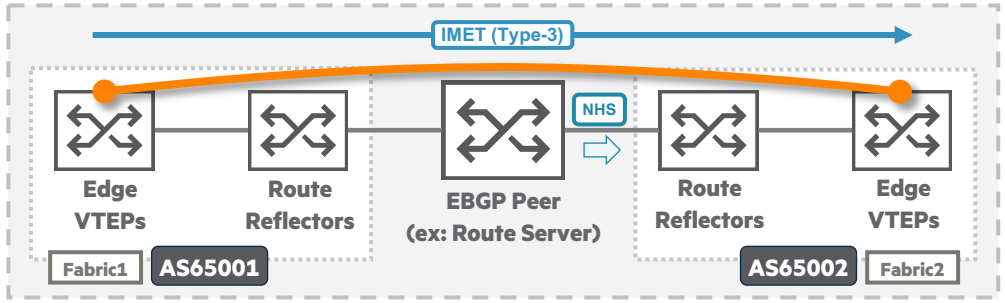
- eBGP must be used between Shared Border and all attached Fabrics
- iBGP must be used within the same Fabric
  - *eBGP within the same Fabric, although configurable for PoC, is not supported in AOS-CX 10.13*
- Shared Border cannot be used as a Route Reflector

- No other changes in comparison to Shared Border use case before AOS-CX 10.13
  - **No impact on Layer 3 capability**

Details
IMET Relay



Tunnels Stitching blocking IMET at Border



Behavior with IMET-Relay enabled.

Statement

Shared Border VTEP, by design, terminates tunnels between Fabrics ("Tunnels Stitching").

- IMET Type-3 EVPN routes must not be exchanged between Fabrics. eBGP EVPN has been modified in AOS-CX 10.13 to natively enforce the filtering of IMET Routes from eBGP Peers to other eBGP Peers (IMET received from eBGP Peers on the Shared Border are still used to construct VXLAN Tunnels for BUM Traffic to all eBGP Edge VTEPs.)
- Shared Border won't forward IMET Routes between Fabrics, to force the tunnel termination on the shared border.

Solution

Allows a Shared Border to act as an IMET Relay, and so to forward those Routes to other VTEPs.

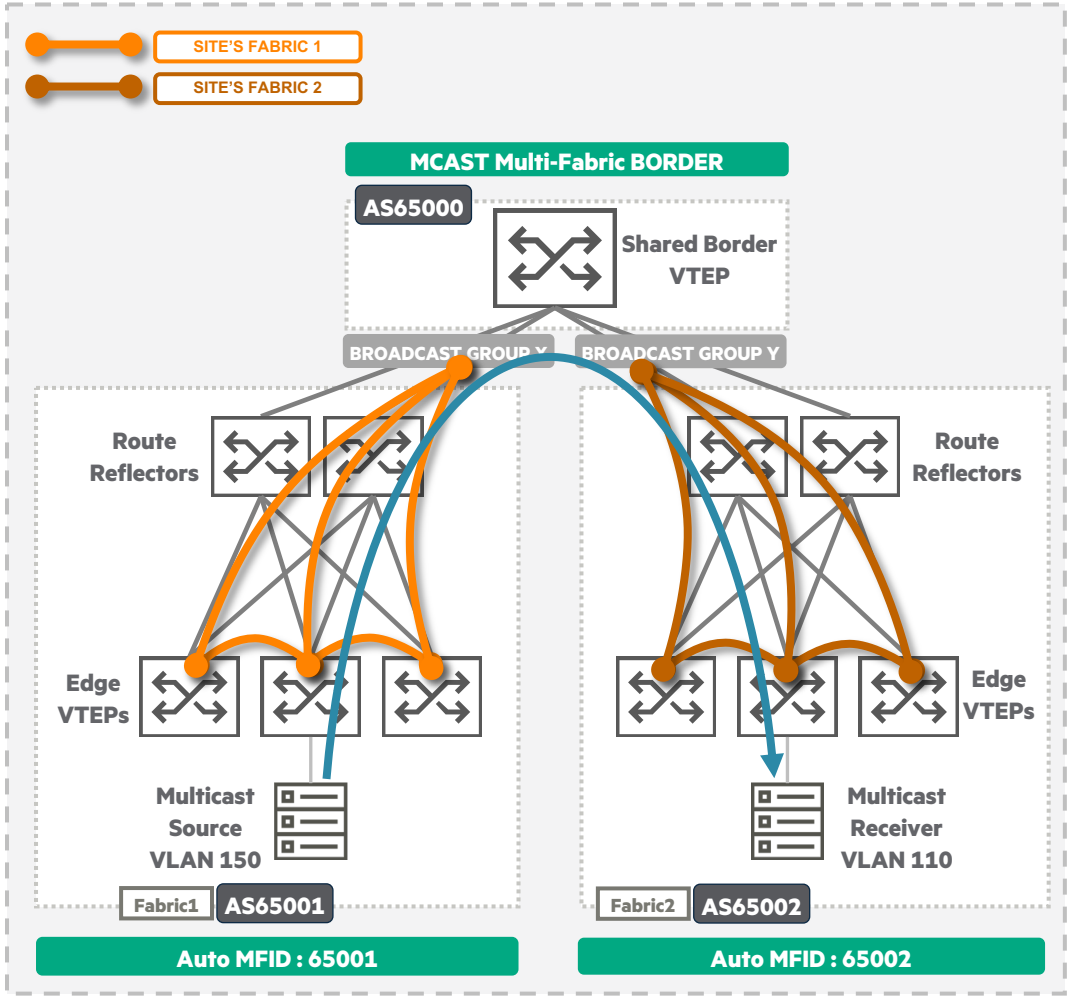
- Configure the "allow-imet-relay" capability in the EVPN configuration context:

```
switch(config)# evpn
switch(config-evpn)# allow-imet-relay
```

Multicast – Layer 3

Details

Multicast Border and Multicast Identifiers



Statement

When Multicast Traffic is received by a VTEP (Here Shared Border), it is not forwarded back to the other VTEPs within the same Fabric (Split horizon rule for Multicast).

Otherwise, other VTEPs within the same Fabric will receive the traffic twice (From the Source VTEP and from the Shared Border)

Solution

Set Broadcast Groups on the Shared Border VTEP and declare this VTEP as a Multicast Multi-Fabric Border.

Use a Multi-Fabric Identifier for each Fabric (MFID).

- The Shared Border will map each PIM neighbor to a MFID.

Broadcast Groups must be configured even if L2 Unicast extension is not required between fabrics

Details

- Multi-Fabric Identifier is used in the PIM Hello packet.
  - It is used by the Shared Border to map the sending PIM Neighbor to a Fabric.
  - All other VTEPs receiving this Identifier won't do anything with it.

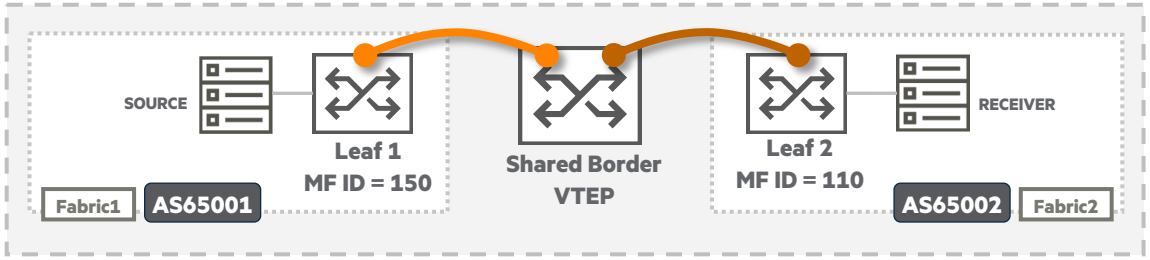
- Broadcast Groups must be set on the Shared Border, to allow Multicast Traffic to be forwarded between Fabrics

- A specific command must be set on the Shared Border to determine it's a Multicast Border Router

- Sources and Receivers can be directly attached to the Shared Border.
- RP and BSR can be external to the Fabrics
- Orphan Ports on Shared Border are supported

Details

Multicast Fabric ID



- Multicast Fabric Identifier is used in the PIM Hello packet.
  - PIM Hello Option 65001 (option reserved for Private Use*) is used to exchange the fabric identifier

```
> User Datagram Protocol, Src Port: 47465, Dst Port: 4789
> Virtual eXtensible Local Area Network
> Ethernet II, Src: ArubaaHe_bc:79:80 (64:e8:81:bc:79:80), Dst: IPv4mcast_0d (01:00:5e:00:00:0d)
> Internet Protocol Version 4, Src: 10.140.254.7, Dst: 224.0.0.13
v Protocol Independent Multicast
    0010 .... = Version: 2
    .... 0000 = Type: Hello (0)
    Reserved byte(s): 00
    Checksum: 0xc0ab [correct]
    [Checksum Status: Good]
  v PIM Options: 5
    > Option 1: Hold Time: 210
    > Option 2: LAN Prune Delay: T = 1, Propagation Delay = 500ms, Override Interval = 2500ms
    > Option 19: DR Priority: 1
    > Option 20: Generation ID: 134122506
    v Option 65001: Address list, old implementation
        Type: 65001
        Length: 8
        old Address List (65001)

0000  b8 d4 e7 d5 31 00 f8 60  f0 d4 0d 00 08 00 45 00   ····1··` ......E·
0010  00 74 1c eb 40 00 3f 11  0d 6d 0a 8c fe 07 0a 8c   ·t··@·?· ·m......
0020  fe 01 b9 69 12 b5 00 60  31 6d 08 00 00 00 00 4e   ···i···` 1m·····N
0030  21 00 01 00 5e 00 00 0d  64 e8 81 bc 79 80 08 00   !···^··· d···y···
0040  45 00 00 42 0f 62 00 00  01 67 c1 52 0a 8c fe 07   E··B·b·· ·g·R····
0050  e0 00 00 0d 20 00 c0 ab  00 01 00 02 00 d2 00[illegible]  ···· ··· ........
0060  00 04 81 f4 09 c4 00 13  00 04 00 00 00[illegible] 00 14  ........ ........
0070  00 04 07 fe 8c 0a fd e9  00 08 00 00 00 96 00 00   ........ ........
0080  00 00                                              ··
```

PIM HELLO FROM LEAF 1
MULTICAST FABRIC ID : 96 (HEXA) = 150 (DECIMAL)

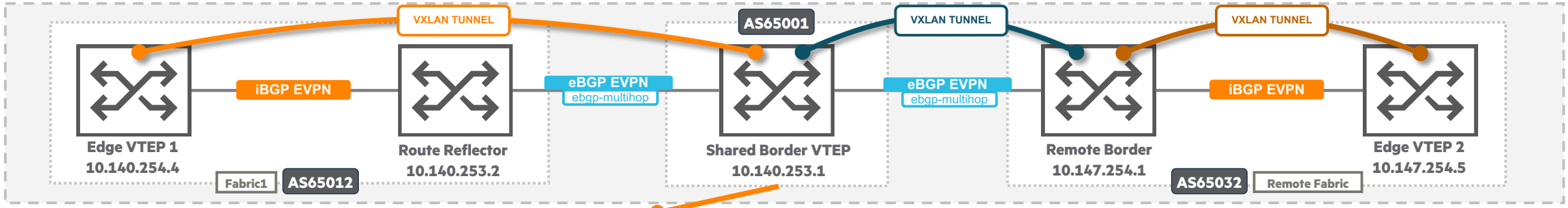
```
> User Datagram Protocol, Src Port: 35238, Dst Port: 4789
> Virtual eXtensible Local Area Network
> Ethernet II, Src: ArubaaHe_99:ef:c0 (88:3a:30:99:ef:c0), Dst: IPv4mcast_0d (01:00:5e:00:00:0d)
> Internet Protocol Version 4, Src: 10.140.254.4, Dst: 224.0.0.13
v Protocol Independent Multicast
    0010 .... = Version: 2
    .... 0000 = Type: Hello (0)
    Reserved byte(s): 00
    Checksum: 0xc3d3 [correct]
    [Checksum Status: Good]
  v PIM Options: 5
    > Option 1: Hold Time: 210
    > Option 2: LAN Prune Delay: T = 1, Propagation Delay = 500ms, Override Interval = 2500ms
    > Option 19: DR Priority: 1
    > Option 20: Generation ID: 83790858
    v Option 65001: Address list, old implementation
        Type: 65001
        Length: 8
        old Address List (65001)

0000  b8 d4 e7 d5 31 00 f8 60  f0 db 8d 00 08 00 45 00   ····1··` ......E·
0010  00 74 d5 00 40 00 3f 11  55 5a 0a 8c fe 04 0a 8c   ·t··@·?· UZ......
0020  fe 01 89 a6 12 b5 00 60  18 c4 08 00 00 00 00 4e   ·******` ******N
0030  21 00 01 00 5e 00 00 0d  88 3a 30 99 ef c0 08 00   !···^··· ·:0·····
0040  45 00 00 42 00 36 00 00  01 67 d0 81 0a 8c fe 04   E··B·6·· ·g******
0050  e0 00 00 0d 20 00 c3 d3  00 01 00 02 00 d2 00[illegible]  ···· ··· ********
0060  00 04 81 f4 09 c4 00 13  00 04 00 00 00[illegible] 00 14  ******** ********
0070  00 04 04 fe 8c 0a fd e9  00 08 00 00 00 6e 00 00   ******** *****n**
0080  00 00                                              ··
```

PIM HELLO FROM LEAF 2
MULTICAST FABRIC ID : 6e (HEXA) = 110 (DECIMAL)

** As per RFC 7761 – Section 4.9.2*

Details

Multicast Fabric Identifier for Remote Fabrics




```
Shared-Border# show ip pim neighbor all-vrfs

< ----- Snipped ----- >

IP Address                  : 10.140.254.4
Interface                   : vni20001
VTEP IP                     : 10.140.254.4
Up Time (HH:MM:SS)          : 7 days 02:49:01
Expire Time (HH:MM:SS)      : 00:00:52
DR Priority                 : 1
Hold Time (HH:MM:SS)        : 00:03:30
Multicast Fabric Identifier: 65012

< ----- Snipped ----- >

IP Address                  : 10.147.254.1
Interface                   : vni20001
VTEP IP                     : 10.147.254.1
Up Time (HH:MM:SS)          : 01:47:33
Expire Time (HH:MM:SS)      : 00:01:02
DR Priority                 : 1
Hold Time (HH:MM:SS)        : 00:03:30
Shared-Border#
```

Remote Border entry.
No line for Multicast Fabric Identifier

```
Shared-Border:/home/admin# ovsdb-client dump PIM_Neighbor address multicast_multifabric_identifier
PIM_Neighbor table
address         multicast_multifabric_identifier
--------------- --------------
"10.140.254.4" 65012
"10.147.254.1" 0
```

Remote Border entry.

Statement

A remote Border, dedicated or shared, will never send a MFID in the PIM Hello Packet.

Solution

For all remote Borders, by design, the receiving Shared Border will automatically set the Multicast Fabric Identifier to 0 for those PIM neighbors.

Details

- Supports both IPv4 and IPv6 in the Overlay
  - Only IPv4 in the underlay

- Distributed Gateway across fabrics is required
  (Routing must be enabled on the Edge VTEPS, as PIM must be used to advertised the MFID)

- For a given VLAN, SVI and Active Gateway can be identical.

- L2VNI Multicast stretching across fabrics is not supported (**No L2 Multicast across attached Fabrics**)
- Only PIM-SM is supported

Configuration

Unicast - VLAN Extension between EVPN-VXLAN Fabrics

Configuration – Unicast – L2 stretched

```
## Route Maps configuration

switch(config)# route-map rmap1 permit seq 10
switch(config-route-map-rmap1-10)# set local-broadcast-group 1
switch(config)# route-map rmap2 permit seq 10
switch(config-route-map-rmap1-10)# set local-broadcast-group 2
switch(config)# route-map rmap3 permit seq 10
switch(config-route-map-rmap1-10)# set local-broadcast-group 3

## Attachment to BGP Peers

switch(config)# router bgp <ASN>
switch(config-bgp)# address-family l2vpn evpn
switch(config-bgp-l2vpn-evpn)# neighbor spine-RR-Fabric1 route-map rmap1 in
switch(config-bgp-l2vpn-evpn)# neighbor spine-RR-Fabric2 route-map rmap2 in
switch(config-bgp-l2vpn-evpn)# neighbor 10.140.254.7 route-map rmap3 in

## ALWAYS : Clear BGP Peerings

switch# clear bgp l2vpn evpn * soft in
- Or -
switch# clear bgp *
```

Actions

- Configure Route Maps to set Broadcast Group ID on a specific set of remote Route Reflectors :
  - Remove the inter-vxlan-briging mode before, otherwise, you won't be able to configure your new route-maps (**Mutually exclusive**)

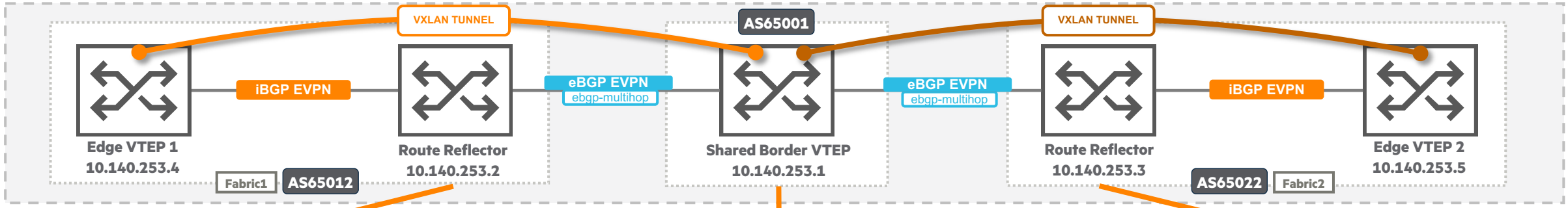
```
Border(config)# route-map rmap1 permit seq 10
Border(config-route-map-rmap1-10)# set local-broadcast-group 1
Dynamic-tunnel-bridging-mode ibgp-ebgp is enabled, Please disable it
```

  - Apply the route-map to the BGP Peers, in the inbound direction (Can be Peer Groups)

- ***IMPORTANT : Always do a soft clear of the BGP sessions after any configuration update.***

Configuration

Simple Example - Unicast




```
router bgp 65012
    bgp router-id 10.140.253.2
    trap-enable
    bgp log-neighbor-changes
    bgp deterministic-med
    bgp always-compare-med
    neighbor 10.140.253.1 remote-as 65001
    neighbor 10.140.253.1 fall-over
    neighbor 10.140.253.1 update-source loopback 0
    neighbor 10.140.253.4 remote-as 65012
    neighbor 10.140.253.4 fall-over
    neighbor 10.140.253.4 update-source loopback 0
    address-family l2vpn evpn
        neighbor 10.140.253.1 next-hop-unchanged
        neighbor 10.140.253.1 send-community both
        neighbor 10.140.253.1 activate
        neighbor 10.140.253.4 route-reflector-client
        neighbor 10.140.253.4 send-community both
        neighbor 10.140.253.4 activate
    exit-address-family
!
```

Route Reflector Fabric 1

```
route-map rmap1 permit seq 10
     set local-broadcast-group 1
route-map rmap2 permit seq 10
     set local-broadcast-group 2
< ------ snipped ------ >
router bgp 65001
    bgp router-id 10.140.253.1
    trap-enable
    bgp log-neighbor-changes
    bgp deterministic-med
    bgp always-compare-med
    neighbor 10.140.253.2 remote-as 65012
    neighbor 10.140.253.2 fall-over
    neighbor 10.140.253.2 update-source loopback 0
    neighbor 10.140.253.3 remote-as 65022
    neighbor 10.140.253.3 fall-over
    neighbor 10.140.253.3 update-source loopback 0
    address-family l2vpn evpn
        neighbor 10.140.253.2 route-map rmap1 in
        neighbor 10.140.253.2 send-community both
        neighbor 10.140.253.2 activate
        neighbor 10.140.253.3 route-map rmap2 in
        neighbor 10.140.253.3 send-community both
        neighbor 10.140.253.3 activate
    exit-address-family
< ------ snipped ------ >
```

Shared Border VTEP

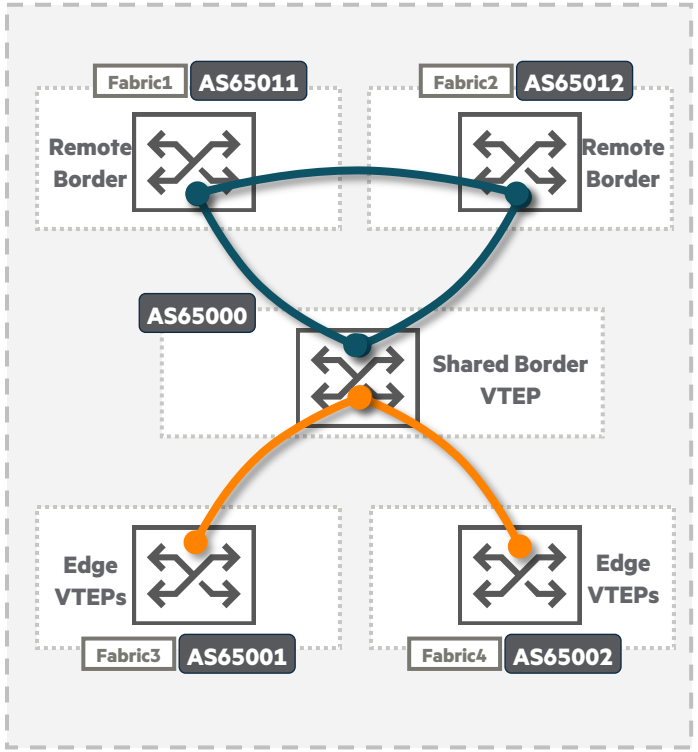
```
router bgp 65022
    bgp router-id 10.140.253.3
    trap-enable
    bgp log-neighbor-changes
    bgp deterministic-med
    bgp always-compare-med
    neighbor 10.140.253.1 remote-as 65001
    neighbor 10.140.253.1 fall-over
    neighbor 10.140.253.1 update-source loopback 0
    neighbor 10.140.253.5 remote-as 65022
    neighbor 10.140.253.5 fall-over
    neighbor 10.140.253.5 update-source loopback 0
    address-family l2vpn evpn
        neighbor 10.140.253.1 next-hop-unchanged
        neighbor 10.140.253.1 send-community both
        neighbor 10.140.253.1 activate
        neighbor 10.140.253.5 route-reflector-client
        neighbor 10.140.253.5 send-community both
        neighbor 10.140.253.5 activate
    exit-address-family
!
```

Route Reflector Fabric 2

BGP configurations on Edge VTEPs, as well as Route Reflectors, remain unchanged

Configuration

Simple Example – With Remote Fabrics dedicated borders



Related Configuration on Shared Border VTEP

```
route-map rmap1 permit seq 10
     set local-broadcast-group 1
route-map rmap2 permit seq 10
     set local-broadcast-group 2
route-map rmap3 permit seq 10
     set local-broadcast-group 3
< ------ snipped ------ >
router bgp 65000
    bgp router-id 10.147.253.1
    trap-enable
< ------ snipped ------ >
address-family l2vpn evpn
        neighbor 10.140.253.1 route-map rmap3 in
        neighbor 10.140.253.1 route-map to-remote out
        neighbor 10.140.253.1 send-community both
        neighbor 10.140.253.1 activate
        neighbor 10.143.253.1 route-map rmap3 in
        neighbor 10.143.253.1 route-map to-remote out
        neighbor 10.143.253.1 send-community both
        neighbor 10.143.253.1 activate
        neighbor 10.147.253.149 route-map rmap1 in
        neighbor 10.147.253.149 send-community both
        neighbor 10.147.253.149 activate
        neighbor 10.147.253.150 route-map rmap2 in
        neighbor 10.147.253.150 send-community both
        neighbor 10.147.253.150 activate
        exit-address-family
```

New route-map

Peering to remote fabrics

Configuration

Clearing BGP Session

AFTER ROUTE MAP APPLICATION

```
Border# show int vxlan vteps detail
Destination                 :  10.140.254.4
Source                      :  10.140.254.1
Origin                      :  evpn
VRF                         :  default
Status                      :  operational
BcastGrpID-Configured       :  NA
BcastGrpID-DataPlane        :  1
Nexthops
========
    IP-ADDRESS        INTERFACE   NEXTHOP-MAC
    ---------------   ----------  --------------------
    10.140.253.101    1/1/3       f8:60:f0:db:8d:00
Destination                 :  10.140.254.5
Source                      :  10.140.254.1
Origin                      :  evpn
VRF                         :  default
Status                      :  operational
BcastGrpID-Configured       :  NA
BcastGrpID-DataPlane        :  1
Nexthops
========
    IP-ADDRESS        INTERFACE   NEXTHOP-MAC
    ---------------   ----------  --------------------
    10.140.253.103    1/1/4       f8:60:f0:d4:0d:00
```

Border# clear bgp *

AFTER CLEARING BGP SESSION

```
Border# show int vxlan vteps detail
Destination                 :  10.140.254.4
Source                      :  10.140.254.1
Origin                      :  evpn
VRF                         :  default
Status                      :  operational
BcastGrpID-Configured       :  1
BcastGrpID-DataPlane        :  4
Nexthops
========
    IP-ADDRESS        INTERFACE   NEXTHOP-MAC
    ---------------   ----------  --------------------
    10.140.253.101    1/1/3       f8:60:f0:db:8d:00
Destination                 :  10.140.254.5
Source                      :  10.140.254.1
Origin                      :  evpn
VRF                         :  default
Status                      :  operational
BcastGrpID-Configured       :  2
BcastGrpID-DataPlane        :  5
Nexthops
========
    IP-ADDRESS        INTERFACE   NEXTHOP-MAC
    ---------------   ----------  --------------------
    10.140.253.103    1/1/4       f8:60:f0:d4:0d:00
```

Configuration
Show commands

Broadcast Groups

```
Switch# show bgp l2vpn evpn neighbors -or- show bgp all neighbors

BGP Neighbor 10.147.253.149 (External)
    Description          :
    Peer-group           :

    Remote Router Id     : 10.147.253.149     Local Router Id     : 10.147.253.1

< ----- Snipped ----- >

    Routemap In          : rmap1
    Routemap Out         :
    ORF type             : Prefix-list
    ORF capability       :

Switch# show interface vxlan vteps detail
Destination            :  10.147.253.159
Source                 :  10.147.254.1
Origin                 :  evpn
VRF                    :  default
Status                 :  operational
BcastGrp-Configured    :  1
BcastGrpID-DataPlane   :  4
Nexthops
========
    IP-ADDRESS       INTERFACE   NEXTHOP-MAC
    ---------------  ----------  --------------------
    10.147.253.53    1/1/4       b8:d4:e7:d5:c0:00

< ----- Snipped ----- >
```

Allow IMET Relay

```
Switch# show run evpn
evpn
    arp-suppression
    redistribute local-svi
    allow-imet-relay
    vlan 110
        rd auto
        route-target export 1:110
        route-target import 1:110
        redistribute host-route
    vlan 200
        rd auto
        route-target export 1:200
        route-target import 1:200
        redistribute host-route
```

Configuration

REST API – GET – ROUTE MAPS

To collect all Broadcast Groups information, 3 different URI can be used :

GET /system/route_maps/{Route_Map.name}/route_map_entries

Parameters

Name	Description
Route_Map.name * required string (path)	Reference Resource: Route_Map Name of the routemap. rmap1

200 Response body


```
{
  "10": {
    "action": "permit",
    "call": null,
    "dampening": {},
    "description": null,
    "exitpolicy": null,
    "goto_target": null,
    "match": {},
    "match_as_path": null,
    "match_community_list": {},
    "match_extcommunity_list": {},
    "match_interface": {},
    "match_ipv4_next_hop_prefix_list": null,
    "match_ipv4_prefix_list": null,
    "match_ipv4_route_source_prefix_list": null,
    "match_ipv6_next_hop_prefix_list": null,
    "match_ipv6_prefix_list": null,
    "match_tag": [],
    "preference": 10,
    "set": {
      "local_broadcast_group_id": 1
```


"/system/route_maps/<rm_name>/route_map_entries"

GET /system/route_maps

Parameters

200 Response body


```
{
  "rmap1": {
    "name": "rmap1",
    "route_map_entries": {
      "10": "/rest/v10.13/system/route_maps/rmap1/route_map_entries/10"
    }
  },
  "rmap2": {
    "name": "rmap2",
    "route_map_entries": {
      "10": "/rest/v10.13/system/route_maps/rmap2/route_map_entries/10"
    }
  },
  "rmap3": {
    "name": "rmap3",
    "route_map_entries": {
      "10": "/rest/v10.13/system/route_maps/rmap3/route_map_entries/10"
    }
  },
  "rmap4": {
    "name": "rmap4",
    "route_map_entries": {
      "10": "/rest/v10.13/system/route_maps/rmap4/route_map_entries/10"
    }
```


"/system/route_maps"

GET /system/vrfs/{VRF.name}/bgp_routers/{BGP_Router.asn}/bgp_neighbors

Parameters

Name	Description
VRF.name * required string (path)	Reference Resource: VRF VRF identifier. Should be alphanumeric. VRF names must be unique. default
BGP_Router.asn * required string (path)	Reference Resource: BGP_Router BGP routers keyed by ASN value. 65000
attributes array[string] (query)	Columns to display. description ebgp_hop_count default_originate_route_map send_community route_maps

200 Response body


```
  },
  "10.141.253.1": {
    "route_maps": {}
  },
  "10.141.253.2": {
    "route_maps": {}
  },
  "10.143.253.1": {
    "route_maps": {
      "l2vpn-evpn,in": "/rest/v10.13/system/route_maps/rmap4",
      "l2vpn-evpn,out": "/rest/v10.13/system/route_maps/to-remote"
    }
  },
  "10.147.253.150": {
    "route_maps": {
      "l2vpn-evpn,in": "/rest/v10.13/system/route_maps/rmap1"
    }
  },
  "10.147.253.150": {
    "route_maps": {
      "l2vpn-evpn,in": "/rest/v10.13/system/route_maps/rmap2"
    }
```


"/system/vrfs/<vrf_name>/bgp_routers/<asn>/bgp_neighbors"

Configuration

REST API – Broadcast Groups - Configuration

You can now use a POST/PUT/PATCH request to configure the Route Map and attach it to a BGP Peer.

```
Shared-Border# show run | begin route-map
route-map rmap1 permit seq 10
     set local-broadcast-group 1
route-map rmap2 permit seq 10
     set local-broadcast-group 2
route-map rmap3 permit seq 10
     set local-broadcast-group 3
route-map to-remote permit seq 10
     match aspath-list local-fabric

Shared-Border# show run bgp
router bgp 65000
< ------ snipped ------ >
  address-family l2vpn evpn
< ------ snipped ------ >
     neighbor 10.143.253.1 send-community both
     neighbor 10.143.253.1 activate
```

Configuration deployment, REST API

```
Shared-Border# show run | begin route-map
route-map rmap1 permit seq 10
     set local-broadcast-group 1
route-map rmap2 permit seq 10
     set local-broadcast-group 2
route-map rmap3 permit seq 10
     set local-broadcast-group 3
route-map rmap4 permit seq 10
     set local-broadcast-group 4
route-map to-remote permit seq 10
     match aspath-list local-fabric

Shared-Border# show run bgp
router bgp 65000
< ------ snipped ------ >
   address-family l2vpn evpn
< ------ snipped ------ >
      neighbor 10.143.253.1 route-map rmap4 in
      neighbor 10.143.253.1 route-map to-remote out
      neighbor 10.143.253.1 send-community both
      neighbor 10.143.253.1 activate
```

```
arnaudlegall@Arnauds-MacBook-Pro CX % python3 shared_border.py
Configuring Shared Border
├── Creating Route Map "rmap4"
├── Creating the Route Map Entry
├── Attaching Route Map to BGP Peer
└── Done
```

There is no clear BGP capability using CX REST API.
Do not forget to do that manually once the configuration is deployed, or to include a SSH command in your workflow.

Configuration
ANSIBLE

Ansible Collection for ArubaOS-CX can be used to configure a Shared Border

```
Shared-Border# show run | begin route-map
route-map rmap1 permit seq 10
     set local-broadcast-group 1
route-map rmap2 permit seq 10
     set local-broadcast-group 2
route-map rmap3 permit seq 10
     set local-broadcast-group 3
route-map to-remote permit seq 10
     match aspath-list local-fabric

Shared-Border# show run bgp
router bgp 65000
< ------ snipped ------ >
  address-family l2vpn evpn
< ------ snipped ------ >
     neighbor 10.143.253.1 send-community both
     neighbor 10.143.253.1 activate
```

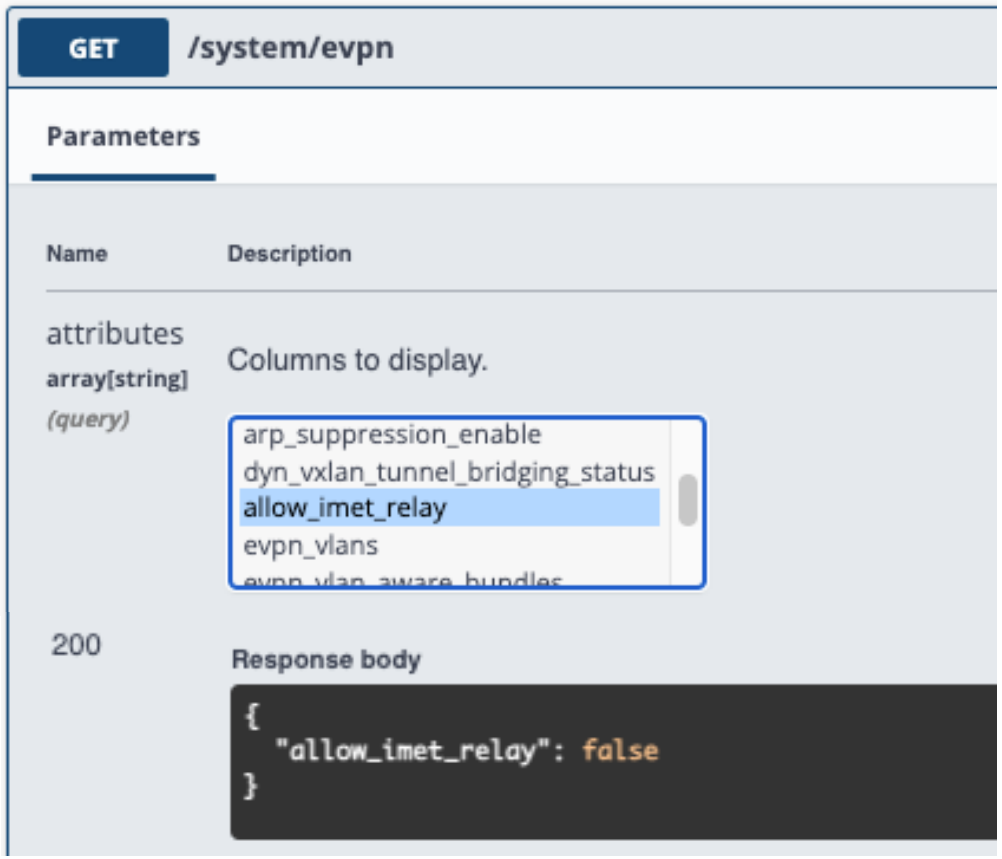
Configuration deployment, based on config template

```
Shared-Border# show run | begin route-map
route-map rmap1 permit seq 10
     set local-broadcast-group 1
route-map rmap2 permit seq 10
     set local-broadcast-group 2
route-map rmap3 permit seq 10
     set local-broadcast-group 3
route-map to-remote permit seq 10
     match aspath-list local-fabric
route-map rmap4 permit seq 10
     set local-broadcast-group 4

Shared-Border# show run bgp
router bgp 65000
< ------ snipped ------ >
   address-family l2vpn evpn
< ------ snipped ------ >
      neighbor 10.143.253.1 route-map rmap4 in
      neighbor 10.143.253.1 route-map to-remote out
      neighbor 10.143.253.1 send-community both
      neighbor 10.143.253.1 activate
```

```
root@85ef4402cecb:/ansible# ansible-playbook -i inventory_shared.yml deploy_shared.yml -e "{'device': 'border'}"

PLAY [border] ***********************************************************************************************

TASK [Generate BG configuration] ****************************************************************************
changed: [border]

TASK [Deploy BG Config] *************************************************************************************
changed: [border]

TASK [Clear BGP Session] ************************************************************************************
ok: [border]

PLAY RECAP **************************************************************************************************
border                     : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Configuration
REST API – ALLOW-IMET-RELAY

To collect Allow IMET Relay status:



"/system/evpn?attributes=allow_imet_relay"

Multicast – Layer 3

Configuration – Multicast – Layer 3

```
## Shared Border

Shared-Border(config)# ip multicast multi-fabric border

## OPTIONAL – Leaf/Access VTEP

# Command
Access-VTEP(config)# ip multicast multi-fabric identifier
 <1-4294967295>

# Fabric 1
Access-VTEP1(config)# ip multicast multi-fabric identifier <identifier1>

# Fabric 2
Access-VTEP2(config)# ip multicast multi-fabric identifier <identifier2>
```

Actions

- **Broadcast Groups must be set for each Fabric.**
  - Hence, remove the inter-vxlan-briging mode before, otherwise, you won't be able to apply your new route-maps (**Mutually exclusive**)
  - *Refer to previous section to see the related configuration.*

- Configure the Shared Border as a Multicast Multi-Fabric Border

Optional

- Configure a Multicast Fabric Identifier on each Leaves / Access VTEPs
  - This Identifier must be unique per Fabric.
  - Different Fabrics cannot share the same Multicast Fabric Identifier

* *The Multicast configuration by itself does not appear here, as it does not change from a traditional Layer 3 Multicast configuration.*

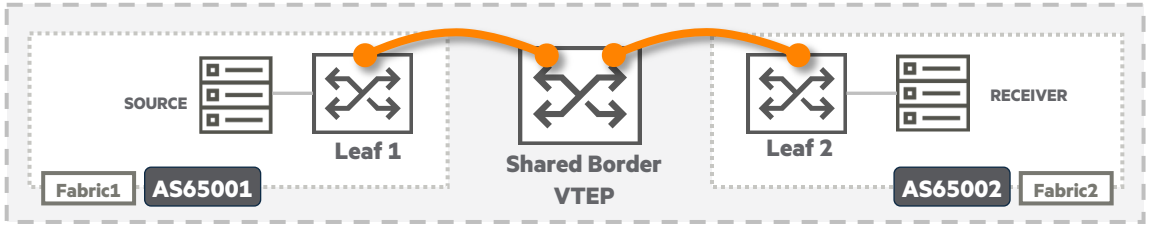
Configuration – Multicast – Layer 3

No MFID Configured

```
# Fabric 1 and Fabric 2
No MFID configuration

# Results on Shared Border
Shared-Border# show ip pim neighbor all-vrfs

< ------ snipped ------ >

IP Address                  : 10.140.254.4
Interface                   : vni20001
< ------ snipped ------ >
Multicast Fabric Identifier: 65001

IP Address                  : 10.140.254.5
Interface                   : vni20001
< ------ snipped ------ >
Multicast Fabric Identifier: 65002
```

MFID Configured

```
# Fabric 1
Access-VTEP1(config)# ip multicast multi-fabric identifier 110

# Fabric 2
Access-VTEP2(config)# ip multicast multi-fabric identifier 150

# Results on Shared Border
Shared-Border# show ip pim neighbor all-vrfs

< ------ snipped ------ >

IP Address                  : 10.140.254.4
Interface                   : vni20001
< ------ snipped ------ >
Multicast Fabric Identifier: 110

IP Address                  : 10.140.254.5
Interface                   : vni20001
< ------ snipped ------ >
Multicast Fabric Identifier: 150
```



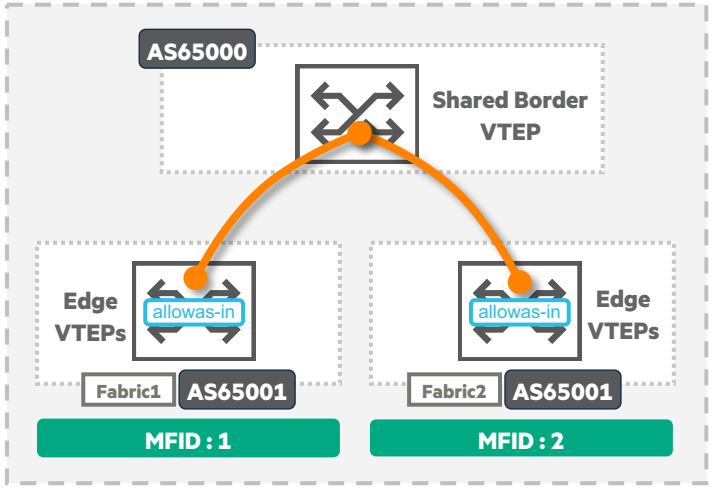
Configuration on Edge VTEPs

Configuration of an Identifier is optional on the VTEPs.

By default, if there is no need to configure the MFID, it will automatically bet set to the local AS Number. **This remains the best practice**.

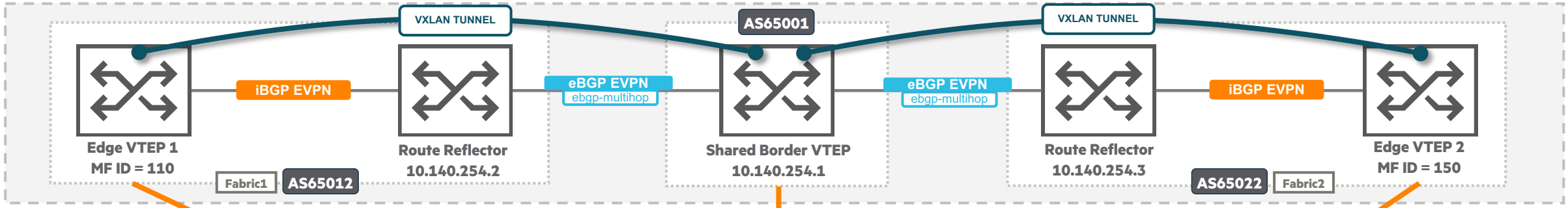
For some use cases, you may need to manually configure the MFID.

Example, when the same AS Number is used by different fabrics :



Configuration

Simple Example - Multicast




```
### Optional

Edge-1# show run
< ------ snipped ------ >
ip multicast multi-fabric identifier 110
```

Edge VTEP 1

```
Shared-Border# show run
< ------ snipped ------ >
ip multicast multi-fabric border
```

Shared Border VTEP

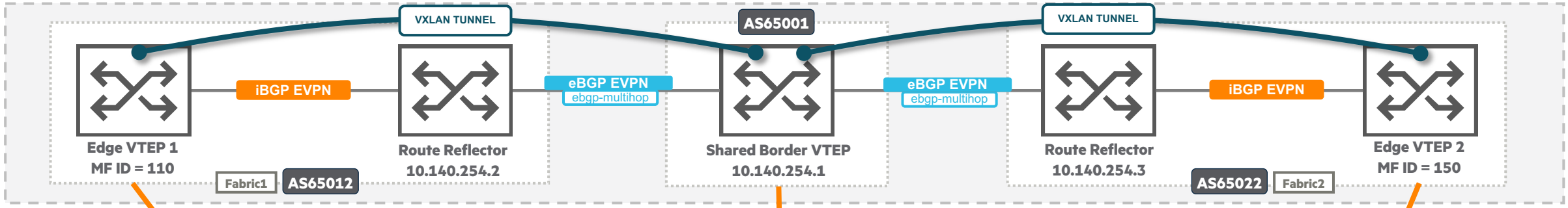
```
### Optional

Edge-2# show run
< ------ snipped ------ >
ip multicast multi-fabric identifier 150
```

Edge VTEP 2

Multicast configurations remain unchanged

Configuration

Simple Example – PIM Configuration




```
Edge-1# show run pim
router pim vrf sense
    enable
interface loopback2
    ip pim-sparse enable
interface vlan110
    ip pim-sparse enable
```

Edge VTEP 1

```
Shared-Border# show run pim
router pim vrf sense
    enable
    rp-candidate source-ip-interface loopback2 group-prefix 224.0.0.0/4
    bsr-candidate source-ip-interface loopback2
```

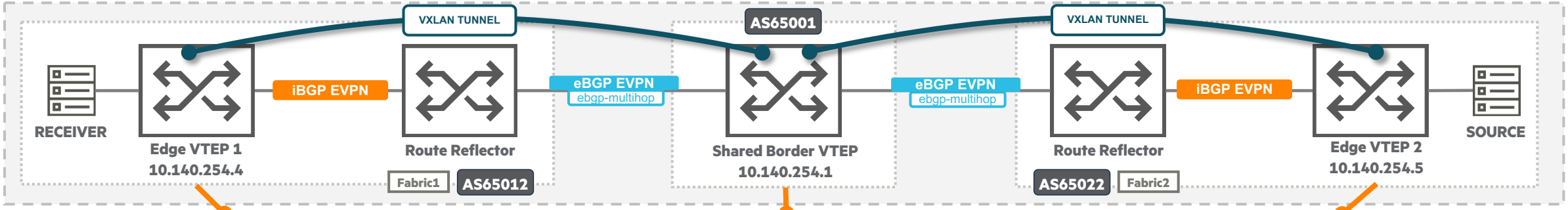
Shared Border VTEP

```
Edge-2# show run pim
router pim vrf sense
    enable
interface loopback2
    ip pim-sparse enable
interface vlan150
    ip pim-sparse enable
```

Edge VTEP 2

Multicast configurations remain unchanged

Configuration
Show commands




```
Edge-1# show ip pim neighbor all-vrfs

PIM Neighbor

< ------ snipped ------ >

VRF                        : sense
Total number of neighbors  : 1

IP Address                 : 10.140.254.4
Interface                  : vni20001
VTEP IP                    : 10.140.254.1
Up Time (HH:MM:SS)         : 00:35:44
Expire Time (HH:MM:SS)     : 00:03:06
DR Priority                : 1
Hold Time (HH:MM:SS)       : 00:03:30
```

Edge VTEP 1

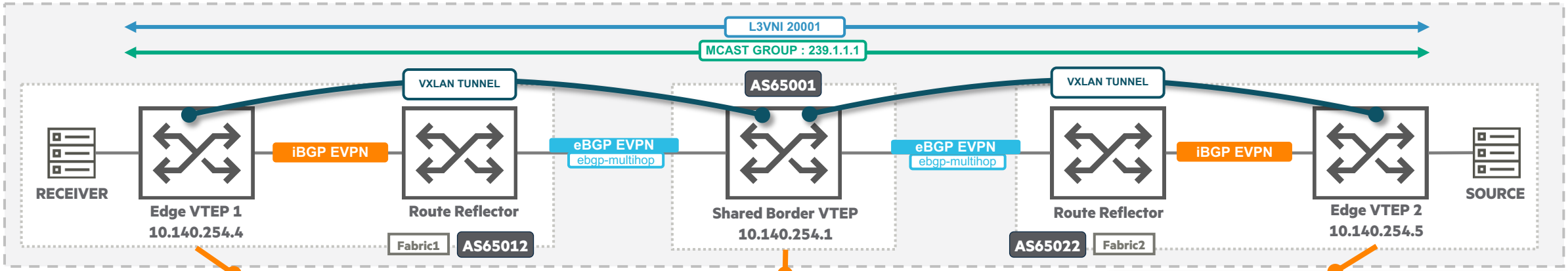
```
Shared-Border# show ip pim neighbor all-vrfs

PIM Neighbor


VRF                        : sense
Total number of neighbors  : 2

IP Address                 : 10.140.254.4
Interface                  : vni20001
VTEP IP                    : 10.140.254.4
Up Time (HH:MM:SS)         : 00:34:41
Expire Time (HH:MM:SS)     : 00:03:24
DR Priority                : 1
Hold Time (HH:MM:SS)       : 00:03:30
Multicast Fabric Identifier: 65012

IP Address                 : 10.140.254.5
Interface                  : vni20001
VTEP IP                    : 10.140.254.5
Up Time (HH:MM:SS)         : 00:32:23
Expire Time (HH:MM:SS)     : 00:03:13
DR Priority                : 1
Hold Time (HH:MM:SS)       : 00:03:30
Multicast Fabric Identifier: 65022
```

Shared Border VTEP

```
Edge-2# show ip pim neighbor all-vrfs

PIM Neighbor

< ------ snipped ------ >

VRF                        : sense
Total number of neighbors  : 1

IP Address                 : 10.140.254.5
Interface                  : vni20001
VTEP IP                    : 10.140.254.1
Up Time (HH:MM:SS)         : 00:34:41
Expire Time (HH:MM:SS)     : 00:03:20
DR Priority                : 1
Hold Time (HH:MM:SS)       : 00:03:30
```

Edge VTEP 2

Configuration
Show commands




```
Edge-1# show ip mroute 239.1.1.1 all-vrfs
IP Multicast Route Entries

VRF : sense
Total number of entries : 1

Group Address                  : 239.1.1.1
Source Address                 : 10.140.150.10
Neighbor                       :
Incoming interface             : vni20001
Multicast Routing Protocol     : PIM-SM
Unicast Routing Protocol       : connected
Metric                         : 0
Metric  Pref                   : 0
Uptime (HH:MM:SS)              : 01:43:47
Downstream Interface
Interface        State
-----------      ----------
vlan110          forwarding
```

Edge VTEP 1

```
Shared-Border# show ip mroute 239.1.1.1 all-vrfs

IP Multicast Route Entries

VRF : sense

Group Address                  : 239.1.1.1
Source Address                 : 10.140.150.10
Neighbor                       : 10.140.254.7
Incoming interface             : vni20001
Intra-Inter Forwarded VTEP(s)  : 10.140.254.4
Multicast Routing Protocol     : PIM-SM
Unicast Routing Protocol       : BGP
Metric                         : 0
Metric  Pref                   : 20
Uptime (HH:MM:SS)              : 00:15:46
```

Shared Border VTEP

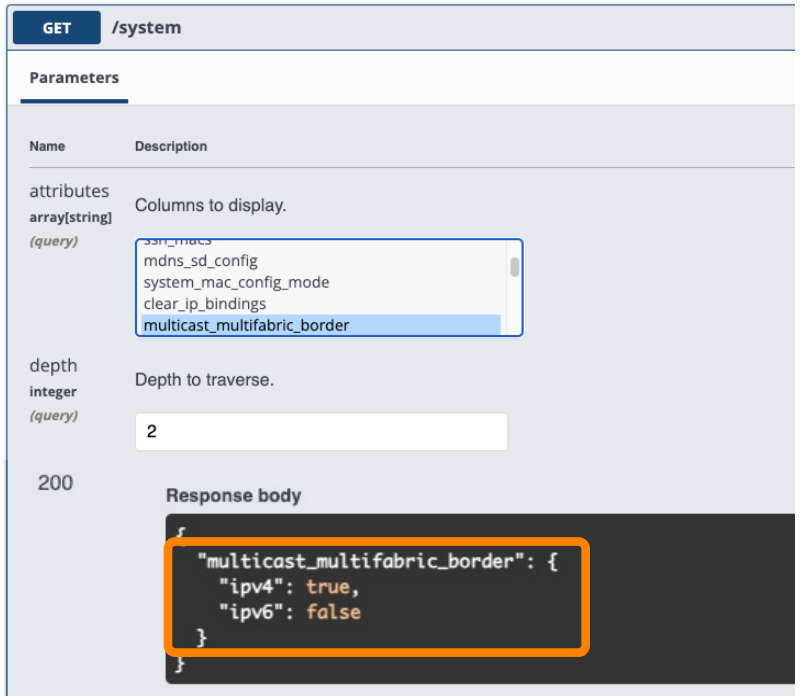
```
Edge-2# show ip mroute 239.1.1.1 all-vrfs
IP Multicast Route Entries

VRF : sense
Total number of entries : 1

Group Address                  : 239.1.1.1
Source Address                 : 10.140.150.10
Neighbor                       :
Incoming interface             : vlan150
Multicast Routing Protocol     : PIM-SM
Unicast Routing Protocol       : connected
Metric                         : 0
Metric  Pref                   : 0
Uptime (HH:MM:SS)              : 01:45:54
Downstream Interface
Interface        State
-----------      ----------
vni20001         forwarding
```

Edge VTEP 2

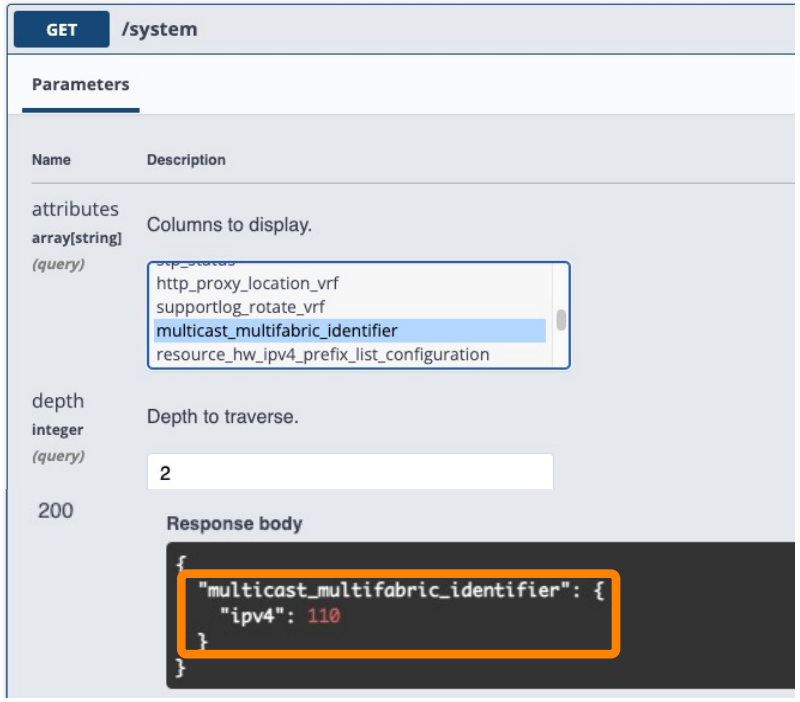
Configuration

REST API – GET – ROUTE MAPS

To collect all Multicast Multi-Fabric information:



SHARED BORDER



ACCESS VTEP / LEAF

URI : '/system"

Configuration

REST API – Broadcast Groups - Configuration

You can now use a POST/PUT/PATCH request to configure the devices.

```
Shared-Border# show run | i border
Shared-Border#
```

```
## Optional
Access-1# show run | i identifier
Access-1#
```

```
## Optional
Access-2# show run | i identifier
Access-2#
```

Configuration deployment, REST API

```
Shared-Border# show run | i border
ip multicast multi-fabric border
Shared-Border#
```

```
## Optional
Access-1# show run | i identifier
ip multicast multi-fabric identifier 110
Access-1#
```

```
## Optional
Access-2# show run | i identifier
ip multicast multi-fabric identifier 150
Access-2#
```

```
arnaudlegall@Arnauds-MacBook-Pro CX % python3 shared_border_multicast.py
Configuring Multicast L3 with Shared Border
├── Enabling MCAST MF on Shared Border
├── Enabling MCAST MF ID on Access VTEPs for Fabric 1 : 110
├── Enabling MCAST MF ID on Access VTEPs for Fabric 2 : 150
└── Done
```

Configuration
ANSIBLE

Ansible Collection for AOS-CX can be used to configure a Shared Border and Fabrics VTEPs

```
Shared-Border# show run | i border
Shared-Border#
```

```
## Optional
Access-1# show run | i identifier
Access-1#
```

```
## Optional
Access-2# show run | i identifier
Access-2#
```

Configuration deployment, based on module "commands"

```
Shared-Border# show run | i border
ip multicast multi-fabric border
Shared-Border#
```

```
## Optional
Access-1# show run | i identifier
ip multicast multi-fabric identifier 110
Access-1#
```

```
## Optional
Access-2# show run | i identifier
ip multicast multi-fabric identifier 150
Access-2#
```

```
root@85ef4402cecb:/ansible# ansible-playbook -i inventory_shared_multicast.yml deploy_shared_multicast.yml

PLAY [all] ***********************************************************************************************************

TASK [Enable MCAST MF on Shared Border] ******************************************************************************
skipping: [Access-1]
skipping: [Access-2]
ok: [border]

TASK [Enable MCAST MF on Access VTEPs] *******************************************************************************
skipping: [border]
ok: [Access-2]
ok: [Access-1]

PLAY RECAP ***********************************************************************************************************
Access-1                   : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
Access-2                   : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
border                     : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Best Practices

Unicast - VLAN Extension between EVPN-VXLAN Fabrics

Best Practices

- For Troubleshoot easiness, configure the Broadcast Group ID for each fabric and do not rely on the default Dataplane ID
- Clearing of BGP Sessions
  - Schedule a maintenance window if you need to implement or update your Broadcast Group configuration
  - Restrict, when possible, to specific peers and Address Families
- If you allow IMET Routes to be relayed, do not configure the device as a VTEP.
  - Otherwise, the remote Leaf will receive 2 different IMET Type-3 EVPN Routes, as the Shared Border will also generate IMET for its local VLAN. It could lead to abnormal behavior.

IMET RELAY ENBLED

```
Leaf2# show bgp l2vpn evpn vni 10110 route-type 3

< ------ snipped ------ >

     Network                                          Nexthop                              Metric     LocPrf    Weight   Path
-----------------------------------------------------------------------------------------------------------------------------
Route Distinguisher: 10.147.253.158:110      (L2VNI 10110)
*>i [3]:[0]:[10.147.253.158]                          10.147.254.1                         0          100        0       65000 65010 ?
Route Distinguisher: 10.147.253.159:110      (L2VNI 10110)
*>  [3]:[0]:[10.147.253.159]                          10.147.253.159                       0          100        0       ?
* i [3]:[0]:[10.147.253.159]                          10.147.253.159                       0          100        0       ?
Route Distinguisher: 10.147.254.1:110      (L2VNI 10110)
*>i [3]:[0]:[10.147.254.1]                            10.147.254.1                         0          100        0       65000 ?

Leaf2# show int vxlan vteps
Source           Destination      Origin       Status                VNI       Routing   VLAN  VRF
-----------------------------------------------------------------------------------------------------
10.147.253.159   10.147.253.158   evpn         operational           10110     disabled  110   --
10.147.253.159   10.147.254.1     evpn         operational           10110     disabled  110   --
10.147.253.159   10.147.254.1     evpn         operational           20001     enabled   --    users
```

2 tunnels are created – 1 to the border and 1 to the remote leaf. This is an incorrect solution.

Multicast – Layer 3

Best Practices

- It is a Best Practice to configure the Shared Border as RP-Candidate/BSR-Candidate

- Multi-Fabric Identifier : Keep the default behavior (Automatic use of the local AS Number)

Reminder:

- Every fabrics attached to Shared Border must have separate Broadcast Group
- Every remote Fabrics (Fully meshed) must be attached to a single Broadcast Group

Troubleshooting

Unicast - VLAN Extension between EVPN-VXLAN Fabrics

Troubleshooting

Make sure the VXLAN EVPN-Based infrastructure is up and running:

- Check the BGP configuration
- Check the EVPN configuration
- Check the configuration of the VXLAN Interface
- Check on the Route Reflector if the Shared Border peering is set with “next-hop-unchanged”

Once done :

- Check if the correct Broadcast Group has been assigned to the correct BGP Peer, in the EVPN Address Family
- Be sure to have cleared the BGP sessions
- In case the Shared Border acts as a VTEP (Tunnel Stitching), be sure the “allow-imet-relay” feature is not enabled.

Multicast – Layer 3

Troubleshooting

Make sure your VXLAN EVPN-Based infrastructure is up and running:

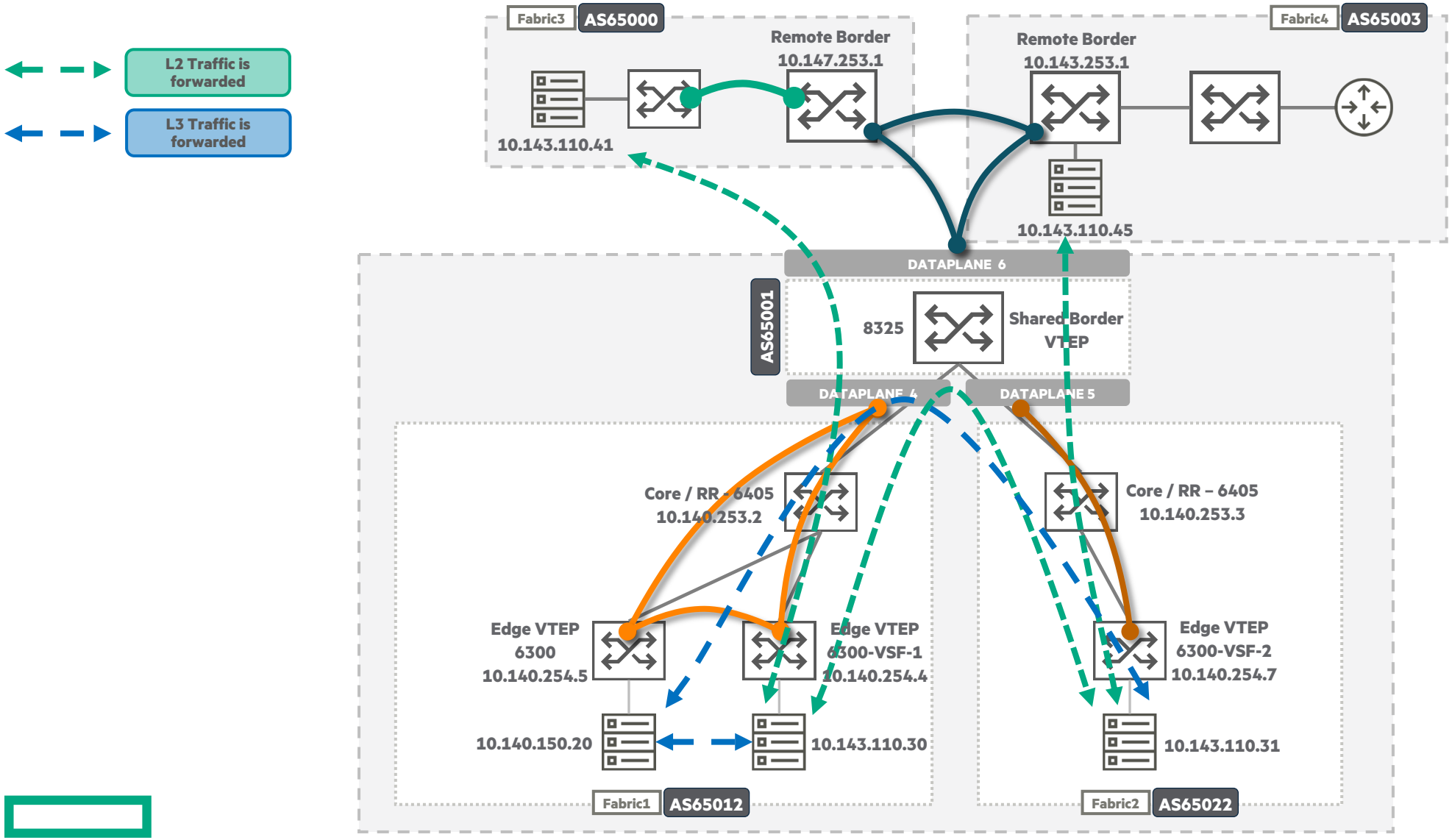
- Check the BGP configuration
- Check the EVPN configuration
- Check the configuration of the VXLAN Interface
- Check on the Route Reflector if the Shared Border peering is set with “next-hop-unchanged”

Once done :

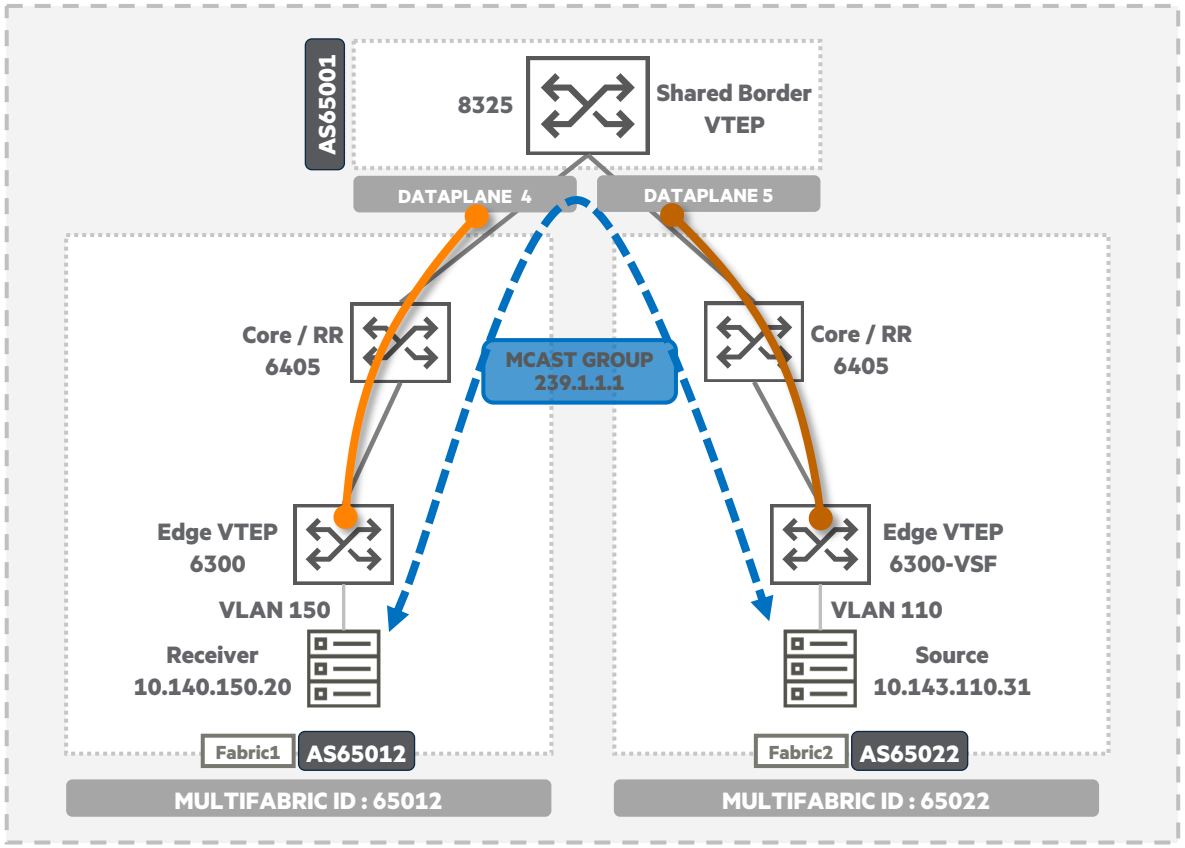
- Check your Multicast configuration (IGMP / IGMP Snooping / PIM)
- Check if the correct Broadcast Group has been assigned to the correct BGP Peer, in the EVPN Address Family
- Check the Multicast Fabric Identifier learned at Shared Border VTEP level
- Check the Multicast Fabric Identifier configuration homogeneity on Edge VTEPs level

Demo

Demo - Stretched Subnet between EVPN-VXLAN Fabrics



Demo - Multicast



Scalability and Performances

Scalability and Performance - Unicast

	Shared Border VTEP			
	8325		8360	
HW profile	Leaf		Agg-Leaf	
Spine / Core	**8325**	**8360**	**8325**	**8360**
VTEPs per Fabric (standalone or VSX logical VTEP pair)	128	64	128	64
Number of Fabrics sharing common core VTEP	4	128	128	128
L3 routes across all VRFs and all sites (including host routes)	23K dual-stack	23K dual-stack	23K dual-stack	23K dual-stack
Overlay hosts (MAC / ARP) within the Fabric on access VTEP	8K dual-stack	8K dual-stack	8K dual-stack	8K dual-stack
Overlay hosts (MAC / ARP) across multiple Fabrics	8K dual-stack	8K dual-stack	8K dual-stack	8K dual-stack
VLANs local to the Fabric	128	64	128	64
Stretched VLANs among all Fabrics	128	64	128	64
VRFs shared among all Fabrics	16	16	16	16

Scalability and Performance - Multicast

	Shared Border VTEP	
	8325	**8360**
VTEPs per Fabric (standalone or VSX logical VTEP pair)	32	32
Number of Fabrics sharing common core VTEP	4	32
IGMP entries	2000	2000
Mroutes	3500	3500
PIM Interfaces (L3VNI)	64	64
L3 VNI/VRF (Each VRF requires 1 x L3 VNI)	32	32

Thank you

aruba_switching_tme@hpe.com

© 2023 Hewlett Packard Enterprise Development LP