OST
Ostschweizer
Fachhochschule

Cloud Infrastructure HS 21

VxLAN eVPN

by Dejan Jovicic and Thomas Kleb

Contents

1. Introduction

In this lab we configured a Nexus 9000 OSv to have VxLAN with two options. First with Flood and Learn using multicast and second the extension using BGP eVPN. To pass the lab we also had to complete the 5 questions / exercises at the end. The Config files (running-config) can be found on our GIT.

The whole lab was set on a network with routers which were preset.

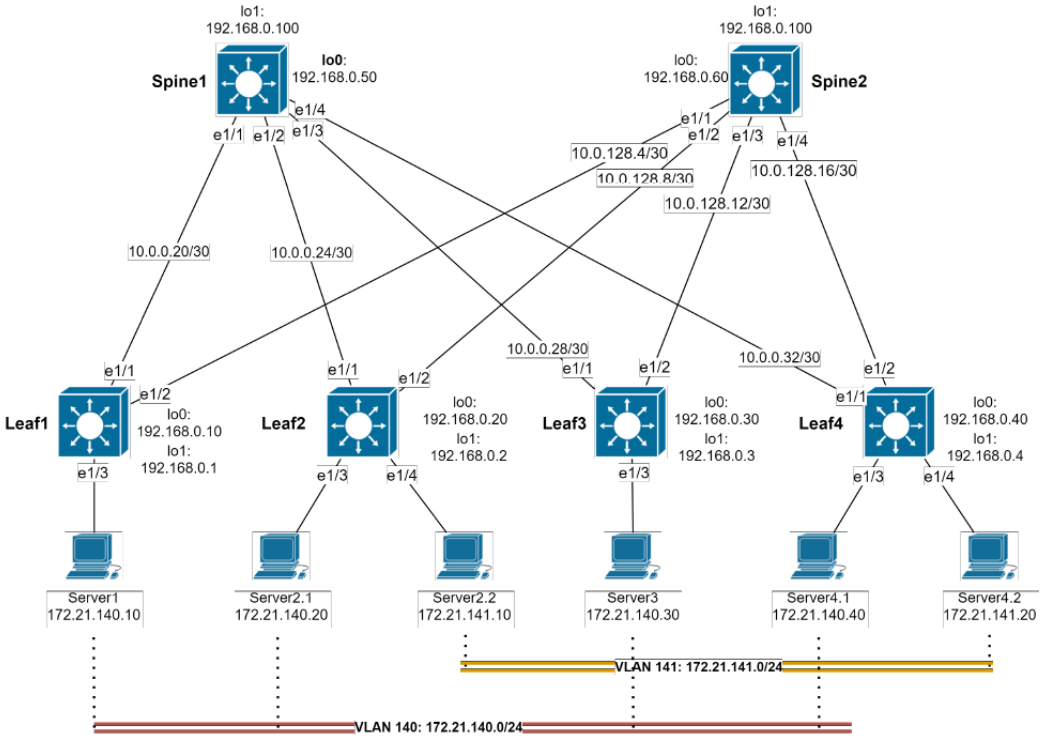


Figure 1 Network Topology

2. Flood and Learn

The goal of this step was to allow the clients (Servers on VLAN 140) to ping between each other. This was achieved by having VTEPs with a given multicast-group. The lab had given us a step-by-step explanation on what to do and our task was to find and implement the right CISCO commands on the routers.

2.1 Configuring Multicast

In this step the goal was to have a rendez-vous point on the spines loopback 1 as well as configuring PIM-ASM on the correct interfaces.

To have pim (Protocol Independent Multicast) working correctly on each of the interfaces we first need to enable “pim sparse mode” on each of the interfaces (Spines and Leaves) that is

connected to a spine. The sparse-mode is the counterpart to dense-mode and doesn't forward any multicast traffic unless there is a receiver in the network that would like to receive said message for a certain multicast-group. But with traffic not flooding everywhere like in dense mode the routers won't know if there is any traffic to be received. To solve this problem, pim sparse-mode uses rendez-vous points (RP). With them each router which receives multicast traffic from a source will forward it to the RP and each router that wants to receive multicast traffic will go to the RP.

To make it work we first had to enable PIM with the command `feature pim`. Then set the RP for each of the Spines and Leaves to the spines' loopback 1 interface and add it to a multicast group: `ip pim rp-address 192.168.0.100 group-list 224.0.0.0/4`.

Now since the RP is configured on both Spines, anycast is needed. On the Nexus 9000 this is set by defining a RP set, which is the set of all routers which would act as RP. To fully enable this function, we must use the second loopback interface "loopback0" on which this set will be defined. Since both Spines have to know the whole set we used the following command on both: `ip pim anycast-rp 192.168.0.100 192.168.0.50` and `.60`.

The first IP is the set RP, and the second IPs are the loopback0 of Spine 1 and Spine 2 respectively. With these commands in place the RP as well as the multicasting is done.

2.2 Creation of VLAN 140

The next step was to create the VLANs and the virtual network identifiers (VNIs) which need to be linked to the VLANs. From the network topology we have given the VLAN 140 and used the VNI of 50140. The implementation of VLAN on the Leaves is rather simple with first enabling it using `feature vn-segment-vlan-based` on all of them. Then adding the VLAN and give it the vn-segment of 50140:

```
vlan 140
    vn-segment 50140
```

To add the vlan on the correct interfaces (Eth 1/3) we had to access them and add the lines: `switchport mode access` and `switchport access vlan 140`.

With these settings in place the VLAN was configured but the clients couldn't ping each other yet. For this to work the next step was needed.

2.3 VxLAN Tunnel Interfaces

For the last step we had to create the VxLAN tunnel interfaces to allow pinging between the clients. To allow this we had to alter the settings on the network virtualization edge (NVE) interface which acts as overlay interface that terminates VxLAN tunnels.

To create and configure the nve first we had to add the feature of network virtualization to the Leaves by adding the line: `feature nv` to each of them. Then for the configurations we added:

```
Interface nve1
    no shutdown
    source-interface loopback1
    member vni 50140
        mcast-group 239.0.0.140
```

Since the source-interface must be a configured loopback interface with a valid /32 IP address we used the given lo1. Then the VxLAN VNI had to be associated with the NVE interface by adding it to 50140. And finally, we had given a new multicast group for this: 239.0.0.140 which was added.

3. eVPN

2.1 Basic iBGP

In this exercise we implemented basic BGP functions into our network but instead of full mesh, route reflection has to be used. This means that instead of every router advertising his information to every other router in the network, each of the Leaves only has to advertise to the spines (dual-homed) which "reflect" to all other iBGP router. All the routers should be in the AS 65000. For the configurations on the Spines each neighborship needs to have IPv4 unicast enabled and `route reflector-client` implemented. `Send community both` enables both standard communities and extended ones. The standard community is used for filtering and tagging in BGP and the extended one carries other information like route-targets for MP-BGP and MPLS VPN. Since loopback 0 is used for peering it has to be given as an "update-source". Here is an example for the Spine 1 neighborship to Leaf 1:

```
router bgp 65000
    address-family ipv4 unicast
    neighbor 192.168.0.10
        remote-as 65000
        update-source loopback0
        address-family ipv4 unicast
            send-community both
            route-reflector-client
```

The configurations for the Leaves are like the ones for the Spines. The only difference is that the route-reflector line isn't needed, and the neighbours are the Spines' IPs.

2.2 VLANs

Configurations for the other VLANs are done the same way as the ones from VLAN 140 (2.2 Creation of VLAN 140) but with the corresponding VNI.

2.3 VRF (Virtual Routing and Forwarding)

VRF allows multiple routing tables on a single router, similar to what VLANs do for switches. In this exercise we use the VLAN 999 with VNI 50999 for L3 routing. The VRF is called "Tenant-1" and linked to the L3 VNI. Then a route distinguisher (RD) is added in auto mode to uniquely identifying a VTEP within an L3VNI. Then in the IPv4 unicast address family the settings for route-target (RT) are configured to both for EVPN and default connections. The RT is used to import and export the IPv4 prefixes:

```
vrf context Tenant-1
    vni 50999
    rd auto
    address-family ipv4 unicast
        route-target both auto
        route-target both auto evpn
```

2.4 SVIs (Switched Virtual Interfaces)

Here we configured the virtual interfaces for each of the VLAN on the Leaves. The steps on how to do it were given with the exercise. An important step was to globally configure the anycast-gateway-mac which we decided to be: 0000.2222.4444. The IP address of the SVI had to be corresponding to the VLAN:

```
Interface Vlan14X
    no shutdown
    vrf member Tenant-1
    no ip redirects
    ip address 172.21.14X.1/24
    fabric forwarding mode anycast-gateway
```

2.5 NVE Interface Adjustments

To adjust the nve interface to the new options we had to add the member vni 50141 of VLAN 141 to the routers which had an Eth1/4 interface in VLAN 141. The corresponding multicast-group is 239.0.0.141. The VRF 50999 is added with an additional attribute “associate-vrf” which is used to identify and separate processing VNIs that are associated with a VRP and used for routing. Additionally, BGP has to be specified as the mechanism for host reachability advertisement. The NVE on a Leaf looks as followed:

```
Interface nve1
    no shutdown
    host-reachability protocol bgp
    source-interface loopback1
    member vni 5014X
        mcast-group 239.0.0.14X
    member vni 50999 associate-vrf
```

2.6 EBGP EVPN

First, we had to adjust the MTU on the physical interfaces because the VXLANs 50-byte header on the packets. The default value is 1500 and therefore a minimal MTU of 1550 needs to be configured (we went with 9192 just to be sure).

Next, on the AS 65000 the line “address-family l2vpn evpn” with “retain route-target all” was added to accept all updates containing any route-target. The final configuration for a neighborship on the Leaves looked as followed (additional route-reflector-client line on the Spines):

```
router bgp 65000
    address-family ipv4 unicast
    address-family l2vpn evpn
        retain route-target all
    neighbor 192.168.0.X
        remote-as 65000
        update-source loopback0
        address-family ipv4 unicast
            send-community both
        address-family l2vpn evpn
            send-community both
```

4. Questions

4.1 First Question

1. Compare the design with Flood&Learn and the design with BGP eVPN. What do you think are the advantages and disadvantages of each control-plane approach (technical and non-technical)? Which solution for the control-plane would you recommend if a company would like to deploy VXLAN in their Datacenter? Justify your answer.

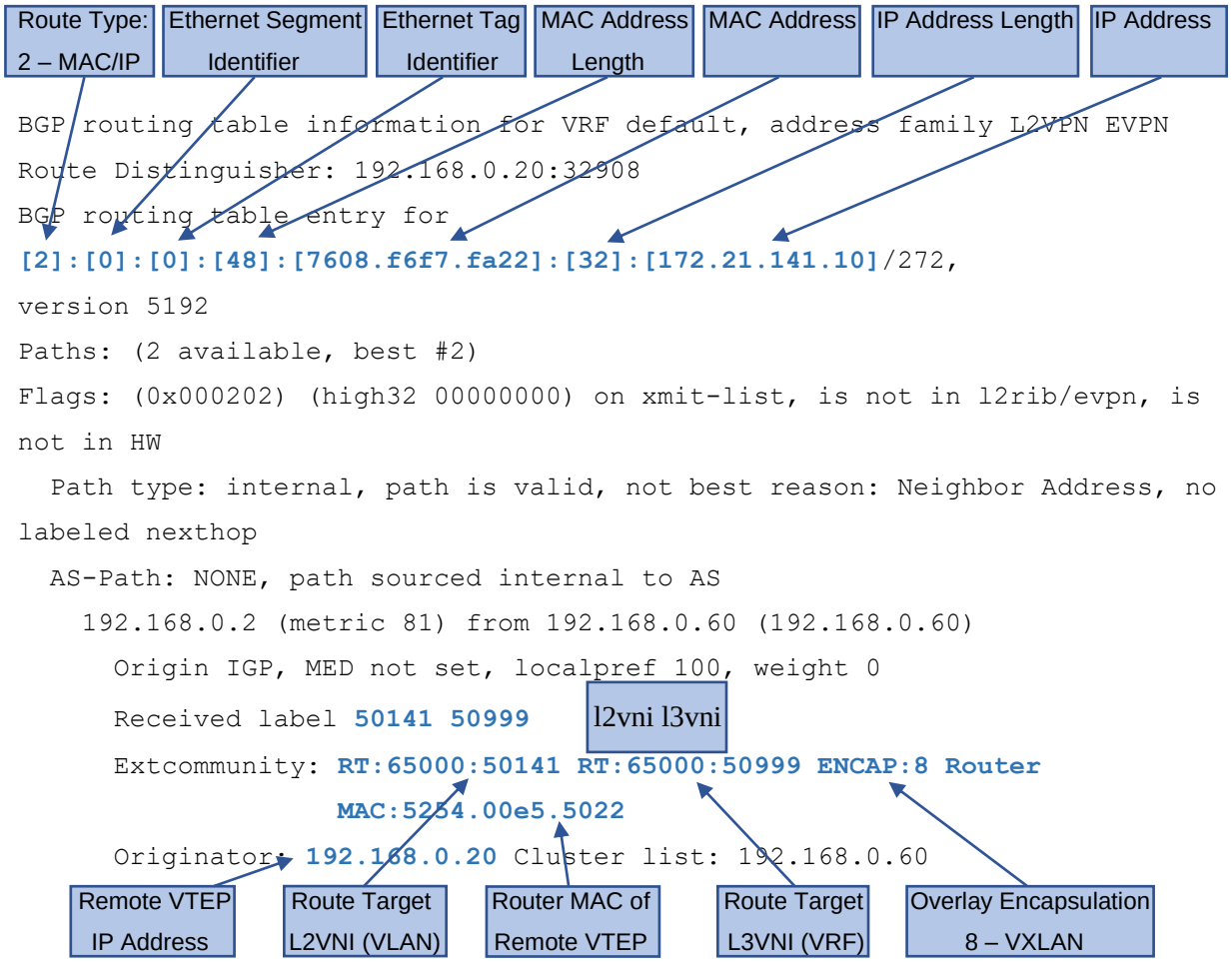
The nature of flooding & learning limits its scalability, there is also no control plane learning of VTEPs, therefore a VTEP could be injected to the network and intercept the traffic.
Why isn't it scalable? If a host wants to send traffic to a host in the same VNI, it will know the IP but not the MAC Address. Therefore, it will send an ARP request to get this information. The Leaf receives the ARP request and forwards it to all VTEPs, to achieve this, it will send the ARP to the associated multicast group. After the VTEP got the ARP, they will forward it to all local ports belonging to the VNI. A host will respond to the ARP. Communicates to the switch, switch to the local VTEP, local VTEP to the original VTEP, original VTEP to the original host. And when the cache expires, this process starts from the beginning. Imagine having 100 VTEPs, the ARP request will be sent to every single one of them, this just isn't scaling well.

BGP is operating in the control plane. Normally, BGP will share its routes, it can also share its MAC addresses and VTEPs reachability information. Every address is learned proactively, therefore flooding never takes places and it isn't needed.
One major advantage to Flood&Learn is, that with BGP neighbour authentication, rogue peers can be prevented. There's also ARP suppression which can be enabled, which means that ARP request won't be flooded to the entire network. As soon as an ARP request reaches the switch (Leaf) , the switch looks at its BGP database, takes the information and sends it back to the host. This means that the design with BGP eVPN is scalable and can even be used with 100 VTEPs.
Another possible advantage is multitenancy. Each L3VNI can be associated with a VRF, this would make multitenancy possible, as each VRF would be configured with a Route Distinguisher to keep it unique.

If we were to have a company, our choice would be to install BGP eVPN. It is better scalable, rogue Leaves wouldn't be possible and we think that in a bigger enterprise it will be easier to keep an overview. Also, in a flood & learn network too much resources would be used for flooding, this problem is basically inexistent in BGP eVPN

4.2 Second Question

From Leaf1 with IP of Leaf 2 Server 2.2: `sh bgp l2vpn evpn 172.21.141.10`




```
Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
  AS-Path: NONE, path sourced internal to AS
    192.168.0.2 (metric 81) from 192.168.0.50 (192.168.0.50)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50141 50999
      Extcommunity: RT:65000:50141 RT:65000:50999 ENCAP:8 Router
MAC:5254.00e5.5022
      Originator: 192.168.0.20 Cluster list: 192.168.0.50
  Path-id 1 not advertised to any peer
```

4.3 Third Question

First, we enabled the BGP ingress replication which forwards Broadcast, Unknown Unicast and Multicast (BUM) traffic to the relevant recipients in a network and is used when IP Multicast underlay network isn't used therefore PIM gets disabled (`no feature pim` on every router). Additionally, we disabled the multicast groups in the member VNI configurations on the NVE interfaces. Then instead of the multicast groups `ingress replication protocol bgp` was added and `global ingress replication protocol bgp` to the outer settings on the NVEs. Finally, the interfaces looked as followed:

```
interface nve1
     no shutdown
     host-reachability protocol bgp
     source-interface loopback1
     global ingress-replication protocol bgp
     member vni 5014X
          ingress-replication protocol bgp
     member vni 50999 associate-vrf
```

To check if the correct type route 3 is now established we once again entered: `sh bgp l2vpn evpn 192.168.0.2` (just the relevant snipped which shows the type. The full output of the command can be found on our GitLab):

```
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.168.0.10:32907    (L2VNI 50140)
BGP routing table entry for [3]:[0]:[32]:[192.168.0.2]/88, version 11581
Paths: (1 available, best #1)
Flags: (0x000012) (high32 00000000) on xmit-list, is in l2rib/evpn, is not
in HW
  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported from 192.168.0.20:32907:[3]:[0]:[32]:[192.168.0.2]/88
  AS-Path: NONE, path sourced internal to AS
    192.168.0.2 (metric 81) from 192.168.0.50 (192.168.0.50)
      Origin IGP, MED not set, localpref 100, weight 0
      Extcommunity: RT:65000:50140 ENCAP:8
      Originator: 192.168.0.20 Cluster list: 192.168.0.50
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 50140, Tunnel Id: 192.168.0.2
```

4.4 Fourth Question

To get a MP-BGP EVPN Route type 5 update we first have to setup another loopback which isn't in the OSPF. Type 5 advertisements / updates happen if there is an outer router / network in another subnet. It advertises the IP prefixes independently of the MAC-advertised routes. We decided on naming it "loopback99" and configured it like that:

```
interface loopback99
    vrf member Tenant-1
    ip address 1.1.1.1/32
```

Then added it to the bgp configurations to allow it in the VRF:

```
router bgp 65000
    vrf Tenant-1
        address-family ipv4 unicast
        network 1.1.1.1/32
```

With the command we used for the second and third question (using the lo99 IP), we now see the type 5 update the new loopback interface causes:

```
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 192.168.0.10:3    (L3VNI 50999)
BGP routing table entry for [5]:[0]:[0]:[32]:[1.1.1.1]/224, version 812
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  Gateway IP: 0.0.0.0
  AS-Path: NONE, path locally originated
    192.168.0.1 (metric 0) from 0.0.0.0 (192.168.0.10)
      Origin IGP, MED not set, localpref 100, weight 32768
      Received label 50999
      Extcommunity: RT:65000:50999 ENCAP:8 Router MAC:5254.0045.f457

  Path-id 1 advertised to peers:
    192.168.0.50       192.168.0.60
```

4.5 Fifth Question

Flood & Learn: Server 1 wants to ping Server 2. To ping to Server 2, Server1 needs to resolve the Server 2 MAC Address first. To do that, it first sends an ARP request with broadcast address (ff:ff:ff:ff:ff:ff) as destination MAC, source MAC is its own MAC. When Leaf1 receives the frame, Leaf1 will learn that the Server 1 MAC Address is XYZ.

As a next step, Leaf1 will now send the broadcast frame to the Rendevouz-Point of the Multicast group attached to the VNI. In our lab that would be VNI 50140 to VLAN 140 with the Multicast group 239.0.0.140. The VTEP "nve1" now creates a Layer 2 entry for the MAC Address and the associated VNI. When all the VTEPs receive the ARP, they cache the IP to MAC information for later. The VTEPs forward the ARP to all ports that belong to the VNI, then VNI to VLAN translation happens. One of the hosts will respond to the ARP, the others will discard the request. The response will be unicast, as it is a single destination. The VTEP from Leaf2 will encapsulate the response and send it back to the original VTEP, which is the one on Leaf1 in our case. He knows he needs to send it there, because he already cached its address when the request came in. The VTEP on Leaf1 receives the response and forwards it to the host, the host then starts normal unicast communication with the destination host.
After the cache has expired, the flooding process will start again from the beginning.

EVPN: All addresses are learned proactively, henceforth there's no need for flooding.
In iBGP, full mesh is needed or route reflectors. We have route reflectors configured at the spines, which is best practice. If a host comes online (for example Server1), he will announce its MAC Address. The Leaf1 will then add the MAC into his local BGP database, this will be sent to its peers as a BGP update. Server1 wants to ping Server2, it will send an ARP request, but now when the ARP request is received at the Leaf1, he will look into his BGP database. The Leaf1 sees the needed info and responds to the host.
Leaf1 L2FWDER component will notice the frame store the MAC Address information to the MAC Address-table. From this table, the information will be installed into L2RIB as a MAC-only entry.

```
Leaf4# show l2route evpn mac evi 140

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

Topology    Mac Address    Prod   Flags         Seq No     Next-Hops
----------- -------------- ------ ------------- ---------- ----------------
140         1ea1.5714.81de BGP    SplRcv        0          192.168.0.2
140         5a36.1ddf.4cf6 BGP    Spl           0          192.168.0.1
140         9eef.bc66.cf27 BGP    SplRcv        0          192.168.0.3
140         d25f.c083.b659 Local  L,            0          Eth1/3
```

Figure 2 L2 Routing Information Base (L2RIB) MAC-Only

Into the L2RIB the MAC-IP information is installed as a MAC-IP entry by the HMM component, which installs the route also into L3RIB. In the screenshot below, we can see the HMM information from the Leaf 4.

```
Leaf4# show fabric forwarding ip local-host-db vrf Tenant-1

HMM host IPv4 routing table information for VRF Tenant-1
Status: *-valid, x-deleted, D-Duplicate, DF-Duplicate and frozen,
        c-cleaned in 00:06:14

    Host                MAC Address        SVI        Flags      Physical Inter
face
*   172.21.140.40/32    d25f.c083.b659     Vlan140    0x420201   Ethernet1/3
*   172.21.141.20/32    d672.fbe9.be08     Vlan141    0x420201   Ethernet1/4
```

Figure 3 VRF Tenant-1 HMM information

In the Figure 4 (below) we can see that the HMM information is produced into the L2RIB, we also see that the information is sent to BGP.

```
Leaf4# sh l2route evpn mac-ip evi 140 detail
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv(D):Del Pending (S):Stale (C):Clear
(Ps):Peer Sync (Ro):Re-Originated (Orp):Orphan
Topology    Mac Address    Host IP         Prod   Flags         Seq No     Next-
Hops
----------- -------------- --------------- ------ ---------- ---------------
140         5a36.1ddf.4cf6 172.21.140.10   BGP    --            0          192.16
8.0.1
140         1ea1.5714.81de 172.21.140.20   BGP    --            0          192.16
8.0.2
140         9eef.bc66.cf27 172.21.140.30   BGP    --            0          192.16
8.0.3
140         d25f.c083.b659 172.21.140.40   HMM    --            0          Local

            Sent To: BGP
```

Figure 4 L2RIB - MAC-IP

MAC-IP information will be installed into the local ARP table.

```
Leaf4# sh ip arp vrf tenant-1

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

IP ARP Table for context Tenant-1
Total number of entries: 2
Address         Age       MAC Address     Interface       Flags
172.21.140.40   00:01:15  d25f.c083.b659  Vlan140
172.21.141.20   00:00:08  d672.fbe9.be08  Vlan141
```

Figure 5 ARP-Table Leaf 4

Into the L3RIB of the VRF, the host route is also installed by the HMM component

```
Leaf4# show ip route 172.21.140.40 vrf tenant-1
IP Route Table for VRF "Tenant-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

172.21.140.40/32, ubest/mbest: 1/0, attached
    *via 172.21.140.40, Vlan140, [190/0], 05:57:44, hmm
```

Figure 6 L3RIB - Server 4.1 route

In the moment, the MAC Address-table, L2RIB, HMM, L2RIB MAC-IP tables, ARP-Table and L3RIB of the Leaves are updated. After the local learning process, the MAC-IP and MAC information are advertised as BGP EVPN Route-Type 2 advertisements to the other Leaves. The MAC-Only advertisement has the Extcommunity RT:65000:50140, whereas the MAC-IP advertisement has the additional RT:65000:50999. The routes will be imported into the corresponding MAC-VRF where the MAC-Only route is installed into the L2RIB and from there to all other Leaf MAC address-tables.

```
Route Distinguisher: 192.168.0.40:32907
BGP routing table entry for [2]:[0]:[0]:[48]:[d25f.c083.b659]:[0]:[0.0.0.0]/216,
 version 1945
Paths: (2 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not i
n HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
  AS-Path: NONE, path sourced internal to AS
    192.168.0.4 (metric 81) from 192.168.0.50 (192.168.0.50)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50140
      Extcommunity: RT:65000:50140 ENCAP:8
      Originator: 192.168.0.40 Cluster list: 192.168.0.50

  Path type: internal, path is valid, not best reason: Neighbor Address, no labe
led nexthop
  AS-Path: NONE, path sourced internal to AS
    192.168.0.4 (metric 81) from 192.168.0.60 (192.168.0.60)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50140
      Extcommunity: RT:65000:50140 ENCAP:8
      Originator: 192.168.0.40 Cluster list: 192.168.0.60

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

Figure 7 BGP table of address-family L2VPN EVPN - MAC-Only

```
BGP routing table entry for [2]:[0]:[0]:[48]:[d25f.c083.b659]:[32]:[172.21.140.4
0]/272, version 130
Paths: (2 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not i
n HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
  AS-Path: NONE, path sourced internal to AS
    192.168.0.4 (metric 81) from 192.168.0.50 (192.168.0.50)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50140 50999
      Extcommunity: RT:65000:50140 RT:65000:50999 ENCAP:8 Router MAC:5254.0048.2
b5f
      Originator: 192.168.0.40 Cluster list: 192.168.0.50

  Path type: internal, path is valid, not best reason: Neighbor Address, no labe
led nexthop
  AS-Path: NONE, path sourced internal to AS
    192.168.0.4 (metric 81) from 192.168.0.60 (192.168.0.60)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50140 50999
      Extcommunity: RT:65000:50140 RT:65000:50999 ENCAP:8 Router MAC:5254.0048.2
b5f
      Originator: 192.168.0.40 Cluster list: 192.168.0.60

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

Figure 8 BGP table of address-family L2VPN EVPN - MAC-IP

```
Route Distinguisher: 192.168.0.20:32907    (L2VNI 50140)
BGP routing table entry for [2]:[0]:[0]:[48]:[d25f.c083.b659]:[0]:[0.0.0.0]/216,
 version 1946
Paths: (1 available, best #1)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.168.0.40:32907:[2]:[0]:[0]:[48]:[d25f.c083.b659]:
[0]:[0.0.0.0]/216
  AS-Path: NONE, path sourced internal to AS
    192.168.0.4 (metric 81) from 192.168.0.50 (192.168.0.50)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50140
      Extcommunity: RT:65000:50140 ENCAP:8
      Originator: 192.168.0.40 Cluster list: 192.168.0.50

  Path-id 1 not advertised to any peer
BGP routing table entry for [2]:[0]:[0]:[48]:[d25f.c083.b659]:[32]:[172.21.140.4
0]/272, version 131
Paths: (1 available, best #1)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop, in rib
             Imported from 192.168.0.40:32907:[2]:[0]:[0]:[48]:[d25f.c083.b659]:
[32]:[172.21.140.40]/272
  AS-Path: NONE, path sourced internal to AS
    192.168.0.4 (metric 81) from 192.168.0.50 (192.168.0.50)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 50140 50999
      Extcommunity: RT:65000:50140 RT:65000:50999 ENCAP:8 Router MAC:5254.0048.2
b5f
      Originator: 192.168.0.40 Cluster list: 192.168.0.50

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

Figure 9 BGP EVPN Instance 50140 BRIB (MAC-Only and MAC-IP)

For Inter-VNI routing, Leaf 2 will install the route from the BGP table into VRF RIB. In the Figure 10 we can see that the route to 172.21.141.20 (Server 4-2, VLAN141), is learned from iBGP. We can see the AS 65000, the VNI 50999 and the tunnel-ID with the encapsulation "VXLAN".

```
Leaf2# sh ip route 172.21.141.20 vrf tenant-1
IP Route Table for VRF "Tenant-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

172.21.141.20/32, ubest/mbest: 1/0
    *via 192.168.0.4%default, [200/0], 07:15:41, bgp-65000, internal, tag 65000
(evpn) segid: 50999 tunnelid: 0xc0a80004 encap: VXLAN
```

Figure 10 BGP entry about Server 4-2 on Leaf 2

This question is sponsored by: https://nwktimes.blogspot.com/2018/11/vxlan-part-xiv-control-plane-operation.html