EVPN Single-Homing Over Segment Routing

Table 1: Feature History

Feature Name	Release Information	Feature Description
EVPN Single-Homing Over Segment Routing	Cisco IOS XE Amsterdam 17.3.1	The EVPN Single-Homing feature utilizes the BGP MPLS-based Ethernet VPN functionality as defined in RFC 7432. That is, to achieve single-homing between a Provider Edge (PE) and a Customer Edge (CE) device. There are three fundamental building blocks for EVPN technology, EVPN Instance (EVI), Ethernet Segment (ES), EVPN BGP routes and extended communities. For EVPN Single-Homing feature, a CE device is attached to a single PE device and has an Ethernet Segment.

The EVPN Single-Homing feature utilizes the functionality defined in RFC 7432 (BGP MPLS-based Ethernet VPN), to achieve single-homing between a Provider Edge (PE) and a Customer Edge (CE) device.

Note

- The EVPN Multi-Homing feature is not supported on the router.
- Associated devices with the Multi-Homing feature connected to the NCS 4200 router are not supported.

• Additional References for EVPN Single-Homing, on page 17

Information about EVPN Single-Homing

Ethernet Multipoint Connectivity

To achieve Ethernet multipoint connectivity, MPLS deployments traditionally rely on Virtual Private LAN Services (VPLS). A VPLS service is built with a full-mesh of pseudowires between PE devices that are part of a Layer 2 broadcast domain. A VPLS PE device performs data-plane MAC learning. For MAC learning, the VPLS PE device uses local interfaces for traffic coming from the access network and uses pseudowires for the traffic coming from the core network.

EVPN Multipoint Solution

EVPN is the next generation of multipoint L2VPN solution that aligns operation principles of L3VPN with Ethernet services. Instead of relying solely on data plane for MAC Address learning, EVPN PE devices signal and learn MAC addresses over the core network using BGP, while still using data plane MAC-learning on the access side. Providers can configure BGP as a common VPN control plane for their ethernet offerings and leverage the advantages of Layer 3 VPN over VPLS. In Cisco IOS XE Fuji 16.8.1, only Single Homing functionality is supported from the feature set defined in RFC 7432.

EVPN Building Blocks

There are three fundamental building blocks for EVPN technology, EVPN Instance (EVI), Ethernet Segment (ES), EVPN BGP routes and extended communities:

- EVI is a VPN connection on a PE router. It is the equivalent of IP VPN Routing and Forwarding (VRF) in Layer 3 VPN. It is also known as MAC-VRF.
- ES is a connection with a customer site (device or network) and is associated with access-facing interfaces. Access-facing interfaces are assigned unique IDs that are referred to as Ethernet Segment Identifiers (ESI). A site can be connected to one or more PEs. The ES connection has the same ESI in each PE connected to the site.
- RFC 7432 defines routes and extended communities to enable EPVN support. In Cisco IOS XE Fuji 16.8.x Software Release, Route Type 2 and Route Type 3 are supported.

  In BGP MPLS-based EVPN, an EVI is configured for every PE device for each customer associated with the PE device. In this case, a customer is any customer edge device that is attached to the PE device. The CE device can be a host, a switch or a router. Each EVI has a unique Route Distinguisher (RD) and one or more Route Targets (RT).

  For EVPN Single-Homing feature, a CE device is attached to a single PE device and has an Ethernet Segment with ESI=0.

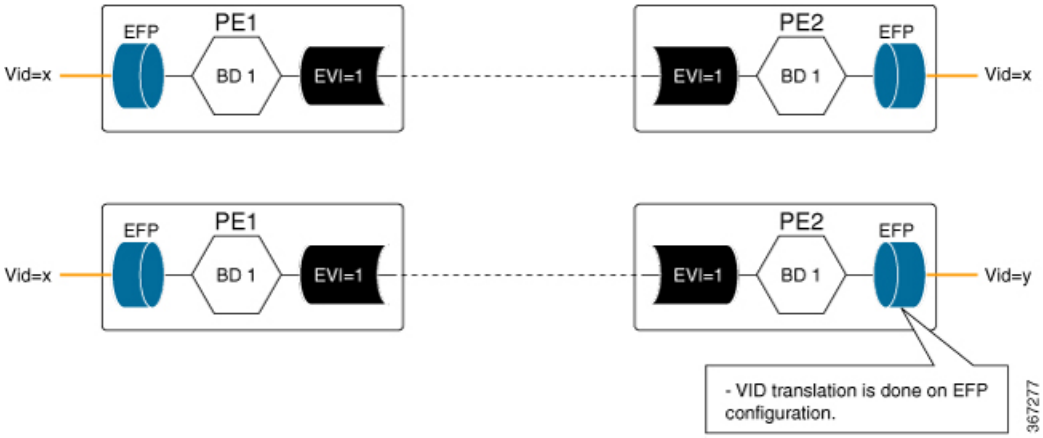
Service Interfaces

The following are types of EVPN VLAN service interfaces:

VLAN-based Service Interface

In VLAN-based service interface, each VLAN is associated to one bridge domain and one EVI.

Figure 1: VLAN-Based Service Interface



For VLAN-based Service Interface, Type 1 Route Distinguisher, a unique number used to distinguish identical routes in different VRFs, is used for EVIs as recommended by the RFC 7432. The Route Distinguishers and Router Targets, which are used to share routes between different VRFs, are autogenerated to ensure unique Route Distinguisher numbers across EVIs.

VLAN Bundle Service Interface

In VLAN Bundle Service Interface, multiple VLANs share the same bridge table.

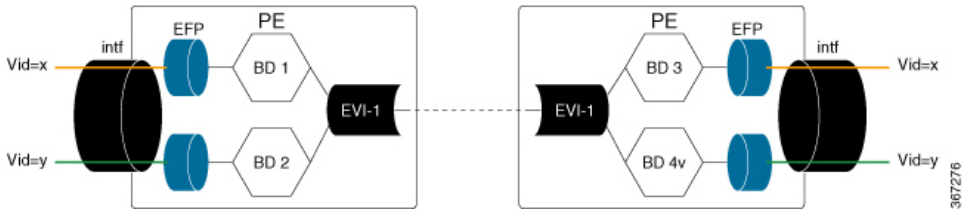
Figure 2: VLAN Bundle Service Interface

Each EVPN instance corresponds to multiple broadcast domains maintained in a single bridge table per MAC-VRF. For VLAN Bundle Service Interface service to work, MAC addresses must be unique across all VLANs for an EVI.

VLAN-Aware Bundle Service Interface

For VLAN-aware Bundle Service Interface, each VLAN is associated with one bridge domain, but there can be multiple bridge domains associated with one EVI.

Figure 3: VLAN-Aware Bundle Service Interface



An EVPN instance consists of multiple broadcast domains where each VLAN has one bridge table. Multiple bridge tables (one per VLAN) are maintained by a single MAC-VRF that corresponds to the EVPN instance.

Route Types

For EVPN Single-Homing feature, Route Type 2 and Route Type 3 are supported, as defined by RFC 7432.

Route Type 2 — MAC and IP Advertisement Route

Type 2 Routes are used to advertise MAC addresses and their associated IP addresses. When a PE router learns the MAC address of a CE device that is connected to it locally, or a MAC address of a device behind the CE device, a MAC and an IP advertisement route is created.

The following table describes the header format for the MAC and IP Advertisement Route packet:

Table 2: Header format for the MAC and IP Advertisement Route packet

Field	Value	Length (Octets)
Route Type	0x02	1
Length	Variable	1
EVI RD	Type 1 (IPv4 address) RD unique across all EVIs on the PE	8
ESI	Ethernet Segment Identifier	10
Ethernet Tag	0 or valid Ethernet Tag	4
MAC Addr Len	48	1
MAC Address	Valid MAC address	6
IP Addr Length	IP address length in bits: 0, 32 or 128	1
IP Address	Optional IP address	0 or 4 or 16
Label1	Valid downstream assigned label to perform forwarding to a CE device based on the destination MAC address	3
Label2	Specifies a second label	0-3
EVI RT	Type 0 (2byteAS) route target	8

Note

- MAC Address field is populated with the CE address.
- IP address field is optional with IP Address length set to 0 bits.
- For EVPN Single-Homing feature, ESI value is always set to 0.
- In the Label field (Label1, Label2), Per-BD or Per-CE labels can be assigned.
  - Per-BD is used when PE advertises a single label for all MAC addresses learned in a given bridge domain.
  - Per-CE label assigns a separate label to each access port in the bridge domain.

Route Type 3 — Inclusive Multicast Ethernet Tag Route

Type 3 routes are used for transporting Broadcast, Unknown Unicast, and Multicast (BUM) traffic to other PE devices across a given EVPN network instance.

The following tables describes the header format for Type 3 routes:

Table 3: Header Format for Type 3 Route Packets

Field	Value	Length (Octets)
Route Type	0x03	1
Length	26 or 38	1
EVI RD	Type 1 (IPv4Addr) RD unique across all EVIs on the PE	8
Ethernet Tag	0 or valid Ethernet Tag	4
IP Addr Length	IP Address Length - 32 bits or 128 bits	1
IP Address	IP Address common for all EVIs (for example, loopback address)	4 or 16
PMSI Tunnel Attr	{1 byte flags = 0}:{1 byte Tunnel Type}:{3 byte label}:{variable length Tunnel Identifier}	Variable
EVI RT	Type 0 (2byteAS) route target	8

The PE devices advertise an Inclusive Multicast Ethernet Tag (IMET) Route for every EVI-Ethernet Tag sequence. The Ethernet Tag is set to 0 for VLAN-based and VLAN-bundling service interfaces. The Ethernet Tag is set to a valid VLAN ID for VLAN-aware bundling service interface.

Type 3 route also carries a Provider Multicast Service Interface (PMSI) Tunnel attribute as specified in RFC 6514 (BGP Encodings and Procedures for MVPNs).

For Ingress Replication, the IMET route is used to advertise the label (in the PMSI Tunnel Attribute) that the other PEs can use to send BUM traffic to the originating PE device.

Prerequisites for EVPN Single-Homing

- EVI and Bridge domains must be in established state with associated MPLS labels.

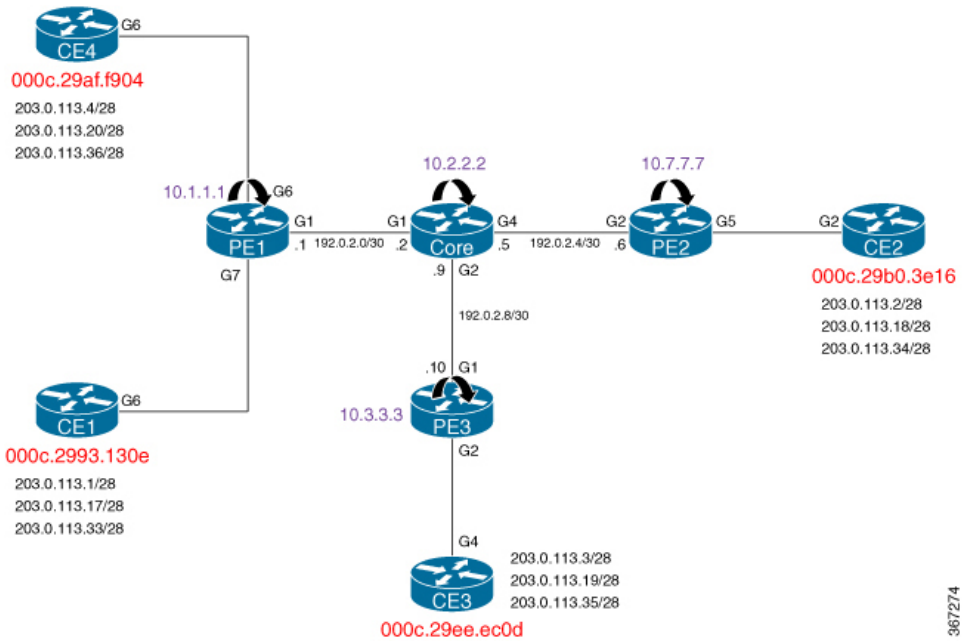
Restrictions for EVPN Single-Homing

- Route Type 1 and Route Type 4 are not supported.
- Per-EVI-based labelling is not supported.
- Maximum number of supported bridge domains is 1600.
- Maximum number of supported EEPs or service instances is 8000.
- Single-Homing feature is not supported with port channel interface between Provider Edge and Customer Edge devices.
- ESI must be all 0s.
- BDI for EVPN bridge domain is not supported.
- EVPN feature does not work with VPLS template enabled.
- EVPN VC statistics not supported with default template.
- SR-TE is not supported with EVPN.
- MACSec with EVPN is not supported.
- EVPN MAC scale is limited to 20000 MACs at up to a maximum rate of 500 PPS.
- EVPN is not supported on TEFP.
- EVPN is not supported with "enable_8k_efp" template.
- Logical port IDs are allotted for Broadcast, Unicast, and Multicast (BUM) traffic and Unicast traffic. So, each EVPN session with a single neighbour utilizes two lportid from the pseudowire bucket and thus, is considered as two pseudowires.
- If the EVPN session neighbour increase, there is one more resource consumed for that EFI interface.
- For VPLS, single lportid per bridge-domain is allocated for each neighbour.
- MAC mobility is *not* supported for EVPN single homing.

How to Configure EVPN Single Homing

Configuring EVPN

Figure 4: EVPN Single Homing



The above figure represents a simple EVPN network. Use the following steps to configure EVPN:

EVPN Configuration

```
enable
    configure terminal
        l2vpn evpn
          replication-type ingress
          router-id Loopback1
          mpls label mode per-ce
         !
        l2vpn evpn instance 10 vlan-based
          route-target both 10:10
          no auto-route-target
         !
        bridge-domain 10
        member evpn-instance 10
          member GigabitEthernet 0/0/1 service-instance 10
        !
        interface GigabitEthernet 0/0/1
          no ip address
          service instance 10 ethernet
           encapsulation dot1q 200
           ¡
          ¡
        !
```

Note In the above example, the **l2vpn evpn instance** command and the associated sub-mode is only required if one or more of the following apply:

- There is per-EVI configuration to be applied (for example, route targets or route distinguished)
- The EVI is VLAN-bundle or VLAN-aware.

If the EVPN instance is not explicitly configured, it is created automatically as a VLAN-based EVI with autogenerated route targets and route distinguisher.

Configuring L2VPN EVPN Globally and EVI on IOS-XE Router

```
l2vpn evpn
 replication-type ingress    ----> Enables ingress replication label
!
l2vpn evpn instance 10 vlan-based ---> Configures Vlan-based EVI 10
!
l2vpn evpn instance 20 vlan-bundle ---→ Configures Vlan-bundled EVI 20
!
l2vpn evpn instance 30 vlan-aware   ---→ Configures Vlan-aware EVI 30
```

Configuring Bridge Domains on IOS-XE Router

```
bridge-domain 10
 mac aging-time 30
 member GigabitEthernet6 service-instance 10  --→ Links SI 10 on interface with Bridge-domain
 10
 member evpn-instance 10 --→ Links EVI 10 with Bridge-domain 10
!
bridge-domain 20
 mac aging-time 30
 member GigabitEthernet6 service-instance 20 --→ Links SI 20 on interface with Bridge-domain
 20
 member evpn-instance 20 --> Links EVI 20 with Bridge-domain 20
!
bridge-domain 30
 mac aging-time 30
 member GigabitEthernet6 service-instance 30 --→ Links SI 30 on interface with Bridge-domain
 30
 member evpn-instance 30 ethernet-tag 30 ---> Links EVI 30 with Bridge-domain 30
```

Configuring Access Interface on a Provider Edge

```
interface GigabitEthernet6
 no ip address
 negotiation auto
 service instance 10 ethernet ----> Enables service instance 10 under the physical interface

  encapsulation dot1q 10
 !
 service instance 20 ethernet ----> Enables service instance 20 under the physical interface

  encapsulation dot1q 20-21
 !
 service instance 30 ethernet ----> Enables service instance 30 under the physical interface

  encapsulation dot1q 30
```

Confguring Native SR for EVPN

```
segment-routing mpls

!

set-attributes

address-family ipv4

sr-label-preferred

exit-address-family

!

global-block 17000 23999

!

connected-prefix-sid-map

address-family ipv4

4.4.4.4/32 index 19 range 1

exit-address-family

!
router ospf 10

router-id 4.4.4.4

nsr

nsf cisco

segment-routing mpls

segment-routing prefix-sid-map advertise-local

fast-reroute per-prefix enable area 0 prefix-priority high

fast-reroute per-prefix enable prefix-priority low

fast-reroute per-prefix ti-lfa

microloop avoidance segment-routing

redistribute connected

network 0.0.0.0 255.255.255.255 area 0

bfd all-interfaces
```

Configuring EVPN Single-Homing

Use the following steps to configure EVPN Single-Homing:

Configuring BGP on Provider Edge Device, PE1

```
enable
configure terminal
 router bgp 100
  bgp router-id 10.1.1.1
  bgp log-neighbor-changes
  bgp graceful-restart
  neighbor 10.2.2.2 remote-as 100
  neighbor 10.2.2.2 update-source Loopback0
 !
 address-family ipv4
  neighbor 10.2.2.2 activate
 exit-address-family
 !
 address-family l2vpn evpn     ----> Enables L2VPN EVPN address family
  neighbor 10.2.2.2 activate
  neighbor 10.2.2.2 send-community both
  neighbor 10.2.2.2 soft-reconfiguration inbound
 exit-address-family
```

Configuring BGP on Route Reflector

```
router bgp 100
 bgp router-id 10.2.2.2
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.3.3.3 update-source Loopback0
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.7.7.7 update-source Loopback0
 !
 address-family ipv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 route-reflector-client
  neighbor 10.7.7.7 activate
  neighbor 10.7.7.7 route-reflector-client
 exit-address-family
 !
 address-family l2vpn evpn      ----> Enables L2vpn evpn address family
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 send-community both
  neighbor 10.1.1.1 route-reflector-client
  neighbor 10.1.1.1 soft-reconfiguration inbound
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 send-community both
  neighbor 10.3.3.3 route-reflector-client
  neighbor 10.3.3.3 soft-reconfiguration inbound
  neighbor 10.7.7.7 activate
  neighbor 10.7.7.7 send-community both
  neighbor 10.7.7.7 route-reflector-client
  neighbor 10.7.7.7 soft-reconfiguration inbound
 exit-address-family
```

Configuring Customer Edge and Provider Edge Interfaces

CE1 configuration

```
interface GigabitEthernet6.10
 encapsulation dot1Q 10
 ip address 203.0.113.1 255.255.255.240
interface GigabitEthernet6.20
 encapsulation dot1Q 20
 ip address 203.0.113.17 255.255.255.240
interface GigabitEthernet6.30
 encapsulation dot1Q 30
 ip address 203.0.113.33 255.255.255.240
```

PE1 Configuration

```
interface GigabitEthernet6
 no ip address
 negotiation auto
 service instance 10 ethernet
  encapsulation dot1q 10
 !
 service instance 20 ethernet
  encapsulation dot1q 20-21
 !
 service instance 30 ethernet
  encapsulation dot1q 30
```

Verification Examples for EVPN Single-Homing

Use the following command to verify that EVI and Bridge domains are in established state and to display associated MPLS labels:

```
show l2vpn evpn evi detail
EVPN instance:    10 (VLAN Based)    ----> VLAN Based EVI
  RD:             10.1.1.1:10 (auto) ----> RD derived from Loopback0 of PE1
  Import-RTs:     100:10
  Export-RTs:     100:10
  Per-EVI Label:  none
  State:          Established    ----> EVI state
  Encapsulation:  mpls
  Bridge Domain:  10
    Ethernet-Tag: 0
    BUM Label:    23             ----> Broadcast/Unknown unicast/Multicast traffic label
    Per-BD Label: 22
    State:        Established     ----> Bridge-domain state
    Pseudoports:
      GigabitEthernet6 service instance 10 ----> Local interface part of bridge-domain
      GigabitEthernet7 service instance 10   ---→ Local interface part of bridge-domain

EVPN instance:    20 (VLAN Bundle) ----> VLAN Bundled EVI
  RD:             10.1.1.1:20 (auto)
  Import-RTs:     100:20
  Export-RTs:     100:20
  Per-EVI Label:  none
  State:          Established
  Encapsulation:  mpls
  Bridge Domain:  20
    Ethernet-Tag: 0
    BUM Label:    20
    Per-BD Label: 21
    State:        Established
    Pseudoports:
      GigabitEthernet6 service instance 20
```

```
      GigabitEthernet7 service instance 20

EVPN instance:    30 (VLAN Aware)  ----> VLAN-Aware EVI
  RD:             10.1.1.1:30 (auto)
  Import-RTs:     100:30
  Export-RTs:     100:30
  Per-EVI Label:  none
  State:          Established
  Encapsulation:  mpls
  Bridge Domain:  30
    Ethernet-Tag: 30
    BUM Label:    18
    Per-BD Label: 19
    State:        Established
    Pseudoports:
      GigabitEthernet6 service instance 30
      GigabitEthernet7 service instance 30
```

Use the following command to see the L2VPN EVPN summary:

```
PE1#show l2vpn evpn summary

L2VPN EVPN

EVPN Instances (excluding point-to-point): 3

VLAN Aware: 1

VLAN Based: 1

VLAN Bundle: 1

Bridge Domains: 4

BGP: ASN 100, address-family l2vpn evpn configured

Router ID: 2.2.2.2

Label Allocation Mode: Per-BD

Global Replication Type: Ingress

MAC Duplication: seconds 180 limit 5

MAC Addresses: 6

Local: 3

Remote: 3

Duplicate: 0

CS1#
```

Use the following command to verify that the bridge domain has learnt the local and remote MAC addresses:

```
PE1#show bridge-domain 10
Bridge-domain 10 (3 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 30 second(s)  ----> MAC aging timer for bridge-domain
    GigabitEthernet6 service instance 10
```

```
 GigabitEthernet7 service instance 10
 EVPN Instance 10
AED MAC address    Policy  Tag       Age  Pseudoport
-   000C.29B0.3E16 forward static_r  0    OCE_PTR:0xe8eb04a0 ---→ Remotely learnt MAC
-   000C.29AF.F904 forward dynamic_c 29   GigabitEthernet6.EFP10 --→ MAC locally learnt

-   000C.2993.130E forward dynamic_c 26   GigabitEthernet7.EFP10
-   000C.29EE.EC0D forward static_r  0    OCE_PTR:0xe8eb0500
```

Note In the above output, MAC addresses with forward dynamic_c tags are locally learned addresses and MAC addresses with forward static_r tags are remote addresses learned through EVPN.

Use the following command to verify that EVPN manager has received the local MACs learned by the bridge domain:

```
PE1# show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop
-------------- ----- ----- ------------------------- ---------- ---------------
000c.2993.130e 10    10    0000.0000.0000.0000.0000  0          Gi7:10
000c.29af.f904 10    10    0000.0000.0000.0000.0000  0          Gi6:10
000c.29b0.3e16 10    10    0000.0000.0000.0000.0000  0          10.7.7.7
000c.29ee.ec0d 10    10    0000.0000.0000.0000.0000  0          10.3.3.3

PE1# show l2vpn evpn mac detail
MAC Address:               000c.2993.130e
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          0000.0000.0000.0000.0000
Ethernet Tag ID:           0
Next Hop(s):               GigabitEthernet7 service instance 10
Label:                     22
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running

MAC Address:               000c.29ee.ec0d
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          0000.0000.0000.0000.0000
Ethernet Tag ID:           0
Next Hop(s):               10.3.3.3
Local Address:             10.1.1.1
Label:                     19
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running
```

Note In the above output, the next hop address of the remote MAC is the address of the provider edge device, if it is learnt remotely or the local interface if MAC address is learnt locally.

Use the following command to verify that Layer 2 Routing Information Base (RIB) has the required the MAC info:

```
PE1# show l2route evpn mac
  EVI       ETag  Prod    Mac Address                    Next Hop(s) Seq Number
----- ---------- ----- -------------- ---------------------------- ----------
   10          0 L2VPN 000C.2993.130E                       Gi7:10          0
   10          0 L2VPN 000C.29AF.F904                       Gi6:10          0
   10          0   BGP 000C.29B0.3E16             L:19 IP:10.7.7.7          0
   10          0   BGP 000C.29EE.EC0D             L:19 IP:10.3.3.3          0
```

Note Remote MACs are learnt through BGP. In the above command output, the producer is BGP and local MACs are learned through Layer 2 VPN.

Use the following command to verify that Layer 2 FIB has received the MAC information from Layer 2 RIB, and bridge-domain and MFI are configured.

```
PE1# show l2fib bridge-domain 10 detail
Bridge Domain : 10
  Reference Count : 18
  Replication ports count : 4
  Unicast Address table size : 4
  IP Multicast Prefix table size : 4

  Flood List Information :
    Olist: Id 9225, Port Count 4

  Port Information :
    Serv Inst: Gi6:10
    Serv Inst: Gi7:10
    EVPN MPLS Encap: pathlist 107
    EVPN MPLS Encap: pathlist 101

  Unicast Address table information :
    Mac: 000c.2993.130e, Adjacency: Serv Inst: Gi7:10
    Mac: 000c.29af.f904, Adjacency: Serv Inst: Gi6:10
    Mac: 000c.29b0.3e16, Adjacency: EVPN MPLS Encap: pathlist 98
    Mac: 000c.29ee.ec0d, Adjacency: EVPN MPLS Encap: pathlist 104

  IP Multicast Prefix table information :
    Source: *, Group: 224.0.0.0/4, IIF:  , Adjacency: Olist: 9226, Ports: 0
    Source: *, Group: 224.0.0.0/24, IIF:  , Adjacency: Olist: 9225, Ports: 4
    Source: *, Group: 224.0.1.39, IIF:  , Adjacency: Olist: 9225, Ports: 4
    Source: *, Group: 224.0.1.40, IIF:  , Adjacency: Olist: 9225, Ports:
```

Use the following command to verify that the information on BGP route type 3 is sent to L2RIB:

```
PE1# show l2route evpn imet
  EVI       ETAG  Prod  Router IP Addr  Type   Label       Tunnel ID
----- ---------- ----- --------------- ----- ------- ---------------
   10          0   BGP         10.3.3.3     6      18         10.3.3.3
   10          0   BGP         10.7.7.7     6      18         10.7.7.7
   10          0 L2VPN         10.1.1.1     6      23         10.1.1.1
```

Use the following command to verify MPLS forwarding:

```
PE1#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
```

```
18          No Label   evpn(mc:bd 30)   305042        none       point2point
19          No Label   evpn(uc:bd 30)   7684          none       point2point
20          No Label   evpn(mc:bd 20)   542588        none       point2point
21          No Label   evpn(uc:bd 20)   13786         none       point2point
22          No Label   evpn(uc:bd 10)   6638          none       point2point
23          No Label   evpn(mc:bd 10)   277740        none       point2point
24          Pop Label  192.0.2.2-A      0             Gi1        192.0.2.2
25          Pop Label  192.0.2.2-A      0             Gi1        192.0.2.2
16001       16001      10.3.3.3/32       0             Gi1        192.0.2.2
16002       Pop Label  10.2.2.2/32       0             Gi1        192.0.2.2
16004       16004      10.7.7.7/32       0             Gi1        192.0.2.2

PE1# show ip bgp l2vpn evpn route-type 2
BGP routing table entry for [2][10.1.1.1:10][0][48][000C2993130E][0][*]/20, version 43
Paths: (1 available, best #1, table evi_10)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (10.1.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      EVPN ESI: 00000000000000000000, Label1 22
      Extended Community: RT:100:10
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [2][10.1.1.1:10][0][48][000C29B03E16][0][*]/20, version 116
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [2][10.7.7.7:10][0][48][000C29B03E16][0][*]/20
 (global)
    10.7.7.7 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 19
      Extended Community: RT:100:10
      Originator: 10.7.7.7, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [2][10.1.1.1:10][0][48][000C29B03E16][0][*]/20, version 116
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [2][10.7.7.7:10][0][48][000C29B03E16][0][*]/20
 (global)
    10.7.7.7 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 19
      Extended Community: RT:100:10
      Originator: 10.7.7.7, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [2][10.1.1.1:10][0][48][000C29EEEC0D][0][*]/20, version 134
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [2][10.3.3.3:10][0][48][000C29EEEC0D][0][*]/20
 (global)
    10.3.3.3 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 19
      Extended Community: RT:100:10
      Originator: 10.3.3.3, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0


PE1# show ip bgp l2vpn evpn route-type 3
BGP routing table entry for [3][10.1.1.1:10][0][32][10.1.1.1]/17, version 41
```

```
Paths: (1 available, best #1, table evi_10)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (10.1.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:100:10
      PMSI Attribute: for EVPN, Flags: 0x0, Tunnel type: 6, length 4, label: 23 (vni 368)
tunnel parameters: 0101 0101
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [3][10.1.1.1:10][0][32][10.3.3.3]/17, version 137
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [3][10.3.3.3:10][0][32][10.3.3.3]/17 (global)

    10.3.3.3 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:10
      Originator: 10.3.3.3, Cluster list: 10.2.2.2
      PMSI Attribute: for EVPN, Flags: 0x0, Tunnel type: 6, length 4, label: 18 (vni 288)
tunnel parameters: 0303 0303
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [3][10.1.1.1:10][0][32][10.7.7.7]/17, version 122
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [3][10.7.7.7:10][0][32][10.7.7.7]/17 (global)

    10.7.7.7 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:10
      Originator: 10.7.7.7, Cluster list: 10.2.2.2
      PMSI Attribute: for EVPN, Flags: 0x0, Tunnel type: 6, length 4, label: 18 (vni 288)
tunnel parameters: 0707 0707
      rx pathid: 0, tx pathid: 0x0
```

Use the following command to verify that segment routing table details:

```
PE1#show segment-routing mpls connected-prefix-sid-map ipv4

               PREFIX_SID_CONN_MAP ALGO_0

    Prefix/masklen   SID Type Range Flags SRGB
        10.0.0.1/32    18 Indx     1         Y

               PREFIX_SID_PROTOCOL_ADV_MAP ALGO_0

    Prefix/masklen   SID Type Range Flags SRGB Source
        10.0.0.1/32    18 Indx     1         Y  IS-IS Level 1 0002.0000.0001
        2.2.2.2/32    19 Indx     1         Y  IS-IS Level 1 0002.0000.0006
        3.3.3.3/32    20 Indx     1         Y  IS-IS Level 1 0002.0000.0002
        4.4.4.4/32    21 Indx     1         Y  IS-IS Level 1 0002.0000.0003
PE1#show segment-routing mpls state
 Segment Routing MPLS State : ENABLED
```

Additional References for EVPN Single-Homing

Standards and RFCs

Standard	Title
RFC 7432	BGP MPLS-Based Ethernet VPN