EVPN Features

This chapter describes how to configure Layer 2 Ethernet VPN (EVPN) features on the router.

- EVPN Overview, on page 1
- EVPN Concepts, on page 2
- EVPN Operation, on page 3
- EVPN Route Types, on page 4
- Configure EVPN L2 Bridging Service, on page 5
- EVPN Software MAC Learning, on page 6
- EVPN Out of Service, on page 14
- EVPN Multiple Services per Ethernet Segment, on page 17
- EVPN Routing Policy, on page 23

### EVPN Overview

Ethernet VPN (EVPN) is a next generation solution that provides Ethernet multipoint services over MPLS networks. EVPN operates in contrast to the existing Virtual Private LAN Service (VPLS) by enabling control-plane based MAC learning in the core. In EVPN, PEs participating in the EVPN instances learn customer MAC routes in control-plane using MP-BGP protocol. Control-plane MAC learning brings a number of benefits that allow EVPN to address the VPLS shortcomings, including support for multi-homing with per-flow load balancing.

EVPN provides the solution for network operators for the following emerging needs in their network:

- Data center interconnect operation (DCI)
- Cloud and services virtualization
- Remove protocols and network simplification
- Integration of L2 and L3 services over the same VPN
- Flexible service and workload placement
- Multi-tenancy with L2 and L3 VPN
- Optimal forwarding and workload mobility
- Fast convergence
- Efficient bandwidth utilization
EVPN Benefits

The EVPN provides the following benefits:

- Integrated Services: Integrated L2 and L3 VPN services, L3VPN-like principles and operational experience for scalability and control, all-active multi-homing and PE load-balancing using ECMP, and enables load balancing of traffic to and from CEs that are multihomed to multiple PEs.

- Network Efficiency: Eliminates flood and learn mechanism, fast-reroute, resiliency, and faster reconvergence when the link to dual-homed server fails, optimized Broadcast, Unknown-unicast, Multicast (BUM) traffic delivery.

- Service Flexibility: MPLS data plane encapsulation, support existing and new services types (E-LAN, E-Line), peer PE auto-discovery, and redundancy group auto-sensing.

EVPN Modes

The following EVPN modes are supported:

- Single-homing - This enables you to connect a customer edge (CE) device to one provider edge (PE) device.

- Multihoming - This enables you to connect a customer edge (CE) device to more than one provider edge (PE) device. Multihoming ensures redundant connectivity. The redundant PE device ensures that there is no traffic disruption when there is a network failure. Following are the types of multihoming:

  - All-Active - In all-active mode all the PEs attached to the particular Ethernet-Segment is allowed to forward traffic to and from that Ethernet Segment.

EVPN Concepts

To implement EVPN features, you need to understand the following concepts:

- Ethernet Segment (ES): An Ethernet segment is a set of Ethernet links that connects a multihomed device. If a multi-homed device or network is connected to two or more PEs through a set of Ethernet links, then that set of links is referred to as an Ethernet segment. The Ethernet segment route is also referred to as Route Type 4. This route is used for designated forwarder (DF) election for BUM traffic.

- Ethernet Segment Identifier (ESI): Ethernet segments are assigned a unique non-zero identifier, which is called an Ethernet Segment Identifier (ESI). ESI represents each Ethernet segment uniquely across the network.

- EVI: The EVPN instance (EVI) is represented by the virtual network identifier (VNI). An EVI represents a VPN on a PE router. It serves the same role of an IP VPN Routing and Forwarding (VRF), and EVIs are assigned import/export Route Targets (RTs). Depending on the service multiplexing behaviors at the User to Network Interface (UNI), all traffic on a port (all-to-one bundling), or traffic on a VLAN (one-to-one mapping), or traffic on a list/range of VLANs (selective bundling) can be mapped to a Bridge Domain (BD). This BD is then associated to an EVI for forwarding towards the MPLS core.

- EAD/ES: Ethernet Auto Discovery Route per ES is also referred to as Route Type 1. This route is used to converge the traffic faster during access failure scenarios. This route has Ethernet Tag of 0xFFFFFFFF.
• EAD/EVI: Ethernet Auto Discovery Route per EVI is also referred to as Route Type 1. This route is used for aliasing and load balancing when the traffic only hashes to one of the switches. This route cannot have Ethernet tag value of 0xFFFFFFFF to differentiate it from the EAD/ES route.

• Aliasing: It is used for load balancing the traffic to all the connected switches for a given Ethernet segment using the Route Type 1 EAD/EVI route. This is done irrespective of the switch where the hosts are actually learned.

• Mass Withdrawal: It is used for fast convergence during the access failure scenarios using the Route Type 1 EAD/ES route.

• DF Election: It is used to prevent forwarding of the loops. Only a single router is allowed to decapsulate and forward the traffic for a given Ethernet Segment.

**EVPN Operation**

At startup, PEs exchange EVPN routes in order to advertise the following:

• **VPN membership**: The PE discovers all remote PE members of a given EVI. In the case of a multicast ingress replication model, this information is used to build the PEs flood list associated with an EVI. BUM labels and unicast labels are exchanged when MAC addresses are learned.

• **Ethernet segment reachability**: In multihoming scenarios, the PE auto-discovers remote PE and their corresponding redundancy mode (all-active or single-active). In case of segment failures, PEs withdraw the routes used at this stage in order to trigger fast convergence by signaling a MAC mass withdrawal on remote PEs.

• **Redundancy Group membership**: PEs connected to the same Ethernet segment (multihoming) automatically discover each other and elect a Designated Forwarder (DF) that is responsible for forwarding Broadcast, Unknown unicast and Multicast (BUM) traffic for a given EVI.

![Figure 1: EVPN Operation](image)

EVPN can operate in single-homing or dual-homing mode. Consider single-homing scenario, when EVPN is enabled on PE, Route Type 3 is advertised where each PE discovers all other member PEs for a given EVPN instance. When an unknown unicast (or BUM) MAC is received on the PE, it is advertised as EVPN Route Type 2 to other PEs. MAC routes are advertised to the other PEs using EVPN Route Type 2. In multihoming scenarios, Route Types 1, 3, and 4 are advertised to discover other PEs and their redundancy modes.
(single-active or all-active). Use of Route Type 1 is to auto-discover other PE which hosts the same CE. The other use of this route type is to fast route unicast traffic away from a broken link between CE and PE. Route Type 4 is used for electing designated forwarder. For instance, consider the topology when customer traffic arrives at the PE, EVPN MAC advertisement routes distribute reachability information over the core for each customer MAC address learned on local Ethernet segments. Each EVPN MAC route announces the customer MAC address and the Ethernet segment associated with the port where the MAC was learned from and its associated MPLS label. This EVPN MPLS label is used later by remote PEs when sending traffic destined to the advertised MAC address.

**EVPN Route Types**

The EVPN network layer reachability information (NLRI) provides different route types.

**Table 1: EVPN Route Types**

<table>
<thead>
<tr>
<th>Route Type</th>
<th>Name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ethernet Auto-Discovery (AD) Route</td>
<td>Few routes are sent per ES, carries the list of EVIs that belong to ES</td>
</tr>
<tr>
<td>2</td>
<td>MAC/IP Advertisement Route</td>
<td>Advertise MAC, address reachability, advertise IP/MAC binding</td>
</tr>
<tr>
<td>3</td>
<td>Inclusive Multicast Ethernet Tag Route</td>
<td>Multicast Tunnel End point discovery</td>
</tr>
<tr>
<td>4</td>
<td>Ethernet Segment Route</td>
<td>Redundancy group discovery, DF election</td>
</tr>
<tr>
<td>5</td>
<td>IP Prefix Route</td>
<td>Advertise IP prefixes.</td>
</tr>
</tbody>
</table>

**Route Type 1: Ethernet Auto-Discovery (AD) Route**

The Ethernet Auto-Discovery (AD) routes are advertised on per EVI and per ESI basis. These routes are sent per ES. They carry the list of EVIs that belong to the ES. The ESI field is set to zero when a CE is single-homed. This route type is used for mass withdrawal of MAC addresses and aliasing for load balancing.

**Route Type 2: MAC/IP Advertisement Route**

These routes are per-VLAN routes, so only PEs that are part of a VNI require these routes. The host's IP and MAC addresses are advertised to the peers within NRLI. The control plane learning of MAC addresses reduces unknown unicast flooding.

**Route Type 3: Inclusive Multicast Ethernet Tag Route**

This route establishes the connection for broadcast, unknown unicast, and multicast (BUM) traffic from a source PE to a remote PE. This route is advertised on per VLAN and per ESI basis.
**Route Type 4: Ethernet Segment Route**

Ethernet segment routes enable to connect a CE device to two or PE devices. ES route enables the discovery of connected PE devices that are connected to the same Ethernet segment.

**Route Type 5: IP Prefix Route**

The IP prefixes are advertised independently of the MAC-advertised routes. With EVPN IRB, host route /32 is advertised using RT-2 and subnet /24 is advertised using RT-5.

---

**Note**

With EVPN IRB, host route /32 are advertised using RT-2 and subnet /24 are advertised using RT-5.

---

**Configure EVPN L2 Bridging Service**

Perform the following steps to configure EVPN L2 bridging service.

---

**Note**

Always ensure to change the label mode from per-prefix to per-VRF label mode. Since L2FIB and VPNv4 route (labels) shares the same resource, BVI ping fails when you exhaust the resources.

---

**Note**

Flooding disable is not supported on EVPN bridge domains.

```bash
/* Configure address family session in BGP */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router#(config)# router bgp 200
RP/0/RSP0/CPU0:router#(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router#(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router#(config-bgp)# neighbor 10.10.10.10
RP/0/RSP0/CPU0:router#(config-bgp)# update-source Loopback 0
RP/0/RSP0/CPU0:router#(config-bgp)# address-family l2vpn evpn

/* Configure EVI and define the corresponding BGP route targets */
Router# configure
Router(config)# evpn
Router(config-evpn)# evi 6005
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# rd 200:50
Router(config-evpn-evi-bgp)# route-target import 100:6005
Router(config-evpn-evi-bgp)# route-target export 100:6005
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# advertise-mac

/* Configure a bridge domain */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group 1
```
Running Configuration

```bash
router bgp 200 bgp
  router-id 209.165.200.227
  address-family 12vpn evpn
  neighbor 10.10.10.10
    remote-as 200 description MPLS-FACING-PEER
    updatesource Loopback0
  addressfamily 12vpn evpn

configure
evpn
evi 6005
  bgp
    rd 200:50
    route-target import 100:6005
    route-target export 100:6005

  advertise-mac

configure
12vpn
  bridge group 1
  bridge-domain 1-1
    interface GigabitEthernet 0/0/0/1
evi 6005

EVPN Software MAC Learning

The MAC addresses learned on one device needs to be learned or distributed on the other devices in a VLAN. EVPN Software MAC Learning feature enables the distribution of the MAC addresses learned on one device to the other devices connected to a network. The MAC addresses are learnt from the remote devices using BGP.

Figure 2: EVPN Software MAC Learning
The above figure illustrates the process of software MAC learning. The following are the steps involved in the process:

1. Traffic comes in on one port in the bridge domain.
2. The source MAC address (AA) is learnt on the PE and is stored as a dynamic MAC entry.
3. The MAC address (AA) is converted into a type-2 BGP route and is sent over BGP to all the remote PEs in the same EVI.
4. The MAC address (AA) is updated on the PE as a remote MAC address.

**Configure EVPN Software MAC Learning**

The following section describes how you can configure EVPN Software MAC Learning:

```plaintext
/* Configure bridge domain. */
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_SH
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface TenGigE 0/0/0/1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface BundleEther 20.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# storm-control broadcast pps 10000 ← Enabling storm-control is optional
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-evi)# commit

/* Configure address family session in BGP. */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router bgp 200
RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 200
RP/0/RSP0/CPU0:router(config-bgp-nbr)# description MPLSFACINGPEER
RP/0/RSP0/CPU0:router(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family 12vpn evpn
```

**Supported Modes for EVPN Software MAC Learning**

The following are the modes in which EVPN Software MAC Learning is supported:

- Single Home Device (SHD) or Single Home Network (SHN)
- Dual Home Device (DHD)—All Active Load Balancing
Single Home Device or Single Home Network Mode

The following section describes how you can configure EVPN Software MAC Learning feature in single home device or single home network (SHD/SHN) mode:

**Figure 3: Single Home Device or Single Home Network Mode**

In the above figure, the PE (PE1) is attached to Ethernet Segment using bundle or physical interfaces. Null Ethernet Segment Identifier (ESI) is used for SHD/SHN.

**Configure EVPN in Single Home Device or Single Home Network Mode**

This section describes how you can configure EVPN Software MAC Learning feature in single home device or single home network mode.

/* Configure bridge domain. */

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_ALL_ACTIVE
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface Bundle-Ether1.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001

/* Configure advertisement of MAC routes. */

RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# evi 2001
RP/0/RSP0/CPU0:router(config-evpn-evi)# advertise-mac

/* Configure address family session in BGP. */

RP/0/RSP0/CPU0:router(config)# configure
RP/0/RSP0/CPU0:router(config)# router bgp 200
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.10.10.10
RP/0/RSP0/CPU0:router(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 200
RP/0/RSP0/CPU0:router(config-bgp-nbr)# description MPLSFACING-PEER
RP/0/RSP0/CPU0:router(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family l2vpn evpn

**Running Configuration**

l2vpn
bridge group EVPN_ALL_ACTIVE
bridge-domain EVPN_2001
interface BundleEther1.2001
evi 2001
! evpn
  evi 2001
  advertise-mac
!
router bgp 200 bgp
router-id 40.40.40.40
address-family l2vpn evpn
neighbor 10.10.10.10
  remote-as 200 description MPLS-FACING-PEER
  updatesource Loopback0
addressfamily l2vpn evpn

Verification
Verify EVPN in single home devices.

RP/0/RSP0/CPU0:router# show evpn ethernet-segment interface Te0/4/0/10 detail

Ethernet Segment Id  Interface     Nexthops
-------------------- ---------- ----------
N/A                  Te0/4/0/10 20.20.20.20

Topology: Operational: SH
Configured: Single-active (AApS) (default)

Dual Home Device—All-Active Load Balancing Mode

The following section describes how you can configure EVPN Software MAC Learning feature in dual home device (DHD) in all-active load balancing mode:

Figure 4: Dual Home Device—All-Active Load Balancing Mode

All-active load-balancing is known as Active/Active per Flow (AApF). In the above figure, identical Ethernet Segment Identifier is used on both EVPN PEs. PEs are attached to Ethernet Segment using bundle interfaces. In the CE, single bundles are configured towards two EVPN PEs. In this mode, the MAC address that is learnt is stored on both PE1 and PE2. Both PE1 and PE2 can forward the traffic within the same EVI.
Configure EVPN Software MAC Learning in Dual Home Device—All-Active Mode

This section describes how you can configure EVPN Software MAC Learning feature in dual home device—all-active mode:

/* Configure bridge domain. */

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_ALL_ACTIVE
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface Bundle-Ether1.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001

/* Configure advertisement of MAC routes. */

RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# evi 2001
RP/0/RSP0/CPU0:router(config-evpn-evi)# advertise-mac
RP/0/RSP0/CPU0:router(config-evpn-evi)# exit
RP/0/RSP0/CPU0:router(config-evpn-evi)# interface bundle-ether1
RP/0/RSP0/CPU0:router(config-evpn-ac)# ethernet-segment
RP/0/RSP0/CPU0:router(config-evpn-ac-es)# identifier type 0 01.11.00.00.00.00.00.01

/* Configure address family session in BGP. */

RP/0/RSP0/CPU0:router(config)# configure
RP/0/RSP0/CPU0:router(config)# router bgp 200
RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router(config-bgp)# address-family 12vpn evpn
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.10.10.10
RP/0/RSP0/CPU0:router(config-bgp)# remote-as 200
RP/0/RSP0/CPU0:router(config-bgp-nbr)# description MPLSFACING-PEER
RP/0/RSP0/CPU0:router(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family 12vpn evpn

/* Configure Link Aggregation Control Protocol (LACP) bundle. */

RP/0/RSP0/CPU0:router(config)# configure
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1.300
RP/0/RSP0/CPU0:router(config-if)# lACP switchover suppress-flaps 300
RP/0/RSP0/CPU0:router(config-if)# exit

/* Configure VLAN Header Rewrite. */

RP/0/RSP0/CPU0:router(config)# configure
RP/0/RSP0/CPU0:router(config)# interface bundle-Ether1.2001 l2transport
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 10
RP/0/RSP0/CPU0:router(config-if)# rewrite ingress tag pop 1 symmetric

Running Configuration

l2vpn
bridge group EVPN_ALL_ACTIVE
bridge-domain EVPN_2001
interface Bundle-Ether1.2001
! evi 2001
!
evpn
  evi 2001
  !
  advertise-mac
  !
  interface bundle-ether1
    ethernet-segment
      identifier type 0 01.11.00.00.00.00.00.01
  !
  !
  router bgp 200
  bgp router-id 209.165.200.227
  address-family l2vpn evpn
  !
  neighbor 10.10.10.10
    remote-as 200
    description MPLS-FACING-PEER
    update-source Loopback0
    address-family l2vpn evpn
  !
  interface Bundle-Ether1
    lacp switchover suppress-flaps 300
    load-interval 30
    !
  interface bundle-Ether1.2001 l2transport
    encapsulation dot1aq 2001
    rewrite ingress tag pop 1 symmetric
  !

Verification

Verify EVPN in dual home devices in All-Active mode.

Note

With the EVPN IRB, the supported label mode is per-VRF.

RP/0/RSP0/CP00:router# show evpn ethernet-segment interface bundle-Ether 1 carvin$

<table>
<thead>
<tr>
<th>Ethernet Segment Id</th>
<th>Interface</th>
<th>Nexthops</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100.211b.fce5.df00.0b00</td>
<td>BE11</td>
<td>10.10.10.10, 209.165.201.1</td>
</tr>
</tbody>
</table>

Topology:

- **Operational**: MHN
- **Configured**: All-active (AApF) (default)
- Primary Services: Auto-selection
- Secondary Services: Auto-selection
- Service Carving Results:
  - Forwarders: 4003
  - Elected: 2002
  - EVI E: 2000, 2002, 36002, 36004, 36006, 36008
  - Not Elected: 2001
  - EVI NE: 2001, 36001, 36003, 36005, 36007, 36009
- MAC Flushing mode: Invalid
- Peering timer: 3 sec [not running]
- Recovery timer: 30 sec [not running]
- Local SHG label: 34251
Verify EVPN Software MAC Learning

Verify the packet drop statistics.

Note

Disable CW configuration if any in EVPN peer nodes, as CW is not supported in EVPN Bridging.

RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name EVPN_2001 details

Bridge group: EVPN_ALL_ACTIVE, bridge-domain: EVPN_2001, id: 1110,
state: up, ShgId: 0, MSTi: 0
List of EVPNs:
  EVPN, state: up
ev1: 2001
  XC ID 0x80000458
Statistics:
packets: received 28907734874 (unicast 9697466652), sent
76882059953
bytes: received 5550285095808 (unicast 1861913597184), sent
14799781851396
MAC move: 0
List of ACs:
  AC: TenGigE0/0/0/1, state is up
  Type VLAN; Num Ranges: 1
  ... Statistics:
packets: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent 45573594908
bytes: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent 8750130222336
MAC move: 0

Verify the EVPN EVI information with the VPN-ID and MAC address filter.

RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 neighbor

Neighbor IP     vpn-id
--------------- ------
209.165.200.225 2001
209.165.201.30  2001

Verify the BGP L2VPN EVPN summary.

RP/0/RSP0/CPU0:router# show bgp l2vpn evpn summary

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>Spk</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>St/PfxRcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>209.165.200.225</td>
<td>0</td>
<td>200</td>
<td>216739</td>
<td>229871</td>
<td>200781341</td>
<td>0</td>
<td>0</td>
<td>3d00h</td>
<td>348032</td>
</tr>
<tr>
<td>209.165.201.30</td>
<td>0</td>
<td>200</td>
<td>6462962</td>
<td>4208831</td>
<td>200781341</td>
<td>10</td>
<td>0</td>
<td>2d22h</td>
<td>35750</td>
</tr>
</tbody>
</table>

Verify the MAC updates to the L2FIB table in a line card.

RP/0/RSP0/CPU0:router# show l2vpn mac mac all location
Verify the MAC updates to the L2FIB table in a route switch processor (RSP).

```console
RP/0/RSP0/CPU0:router# show l2vpn mac mac all location
```

<table>
<thead>
<tr>
<th>Topo ID</th>
<th>Producer</th>
<th>Next Hop(s)</th>
<th>Mac Address</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1112</td>
<td>0/6/CPU0</td>
<td>Te0/0/0/1</td>
<td>00a3.0001.0001</td>
<td></td>
</tr>
</tbody>
</table>

Verify the summary information for the MAC address.

```console
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain EVPN_ALL_ACTIVE:EVPN_2001 mac-address location
```

- **Mac Address**: 0000.2001.5555
  - **Mapped to**: Te0/0/0/2
  - **Learned from/Filtered on**: N/A
  - **LC learned**: N/A
  - **Resync Age/Last Change**: 11 Jan 14:37:22

- **Mac Address**: 00bb.2001.0001
  - **Mapped to**: Te0/0/0/2
  - **Learned from/Filtered on**: N/A
  - **LC learned**: N/A
  - **Resync Age/Last Change**: 11 Jan 14:37:22

- **Mac Address**: 0000.2001.1111
  - **Mapped to**: N/A
  - **Learned from/Filtered on**: N/A

- **Mac Address**: 00a9.2002.0001
  - **Mapped to**: N/A
  - **Learned from/Filtered on**: N/A

Verify the EVPN EVI information with the VPN-ID and MAC address filter.

```console
RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 mac
```

<table>
<thead>
<tr>
<th>EVI</th>
<th>MAC address</th>
<th>IP address</th>
<th>Nexthop</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>00a9.2002.0001</td>
<td>::</td>
<td>10.10.10.10</td>
<td>34226</td>
</tr>
<tr>
<td>2001</td>
<td>00a9.2002.0001</td>
<td>::</td>
<td>209.165.201.30</td>
<td>34202</td>
</tr>
<tr>
<td>2001</td>
<td>0000.2001.5555</td>
<td>20.1.5.55</td>
<td>TenGigE0/0/0/2</td>
<td>34203</td>
</tr>
</tbody>
</table>

```console
RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 mac 00a9.2002.0001 detail
```

- **Ethernet Tag**: 0
- **Multi-paths Resolved**: True <-- aliasing to two remote PE with All-Active load balancing
- **Static**: No
- **Local Ethernet Segment**: N/A
- **Remote Ethernet Segment**: 0100.211b.fce5.df00.0b00
- **Local Sequence Number**: N/A
- **Remote Sequence Number**: 0
- **Local Encapsulation**: N/A
Remote Encapsulation: MPLS

Verify the BGP routes associated with EVPN with bridge-domain filter.

RP/0/RSP0/CPU0:router# show bgp l2vpn evpn bridge-domain EVPN_2001 route-type 2

* > [2][0][48][00bb.2001.0001][0]/104 0.0.0.0
* > [2][0][48][00a9.2002.00be][0]/104 10.10.10.10 100
* i 209.165.201.30 100 0 1

0 i <------ locally learnt MAC
0 i <------ remotely learnt MAC

EVPN Out of Service

The EVPN Out of Service feature enables you to control the state of bundle interfaces that are part of an Ethernet segment that have Link Aggregation Control protocol (LACP) configured. This feature enables you to put a node out of service (OOS) without having to manually shutdown all the bundles on their provider edge (PE).

Use the cost-out command to bring down all the bundle interfaces belonging to an Ethernet VPN (EVPN) Ethernet segment on a node. The Ethernet A-D Ethernet Segment (ES-EAD) routes are withdrawn before shutting down the bundles. The PE signals to the connected customer edge (CE) device to bring down the corresponding bundle member. This steers away traffic from this PE node without traffic disruption. The traffic that is bound for the Ethernet segment from the CE is directed to the peer PE in a multi-homing environment.

EVPN cost-out is supported only on manually configured ESIs.

In the following topology, the CE is connected to PE1 and PE2. When you configure the cost-out command on PE1, all the bundle interfaces on the Ethernet segment are brought down. Also, the corresponding bundle member is brought down on the CE. Hence, the traffic for this Ethernet segment is now sent to PE2 from the CE.

Figure 5: EVPN Out of Service

To bring up the node into service, use no cost-out command. This brings up all the bundle interfaces belonging to EVPN Ethernet segment on the PE and the corresponding bundle members on the CE.
When the node is in cost-out state, adding a new bundle Ethernet segment brings that bundle down. Similarly, removing the bundle Ethernet segment brings that bundle up.

Use `startup-cost-in` command to bring up the node into service after the specified time on reload. The node will cost-out when EVPN is initialized and remain cost-out until the set time. If you execute `evpn no startup-cost-in` command while timer is running, the timer stops and node is cost-in.

The 'cost-out' configuration always takes precedence over the 'startup-cost-in' timer. So, if you reload with both the configurations, cost-out state is controlled by the 'cost-out' configuration and the timer is not relevant. Similarly, if you reload with the startup timer, and configure 'cost-out' while timer is running, the timer is stopped and OOS state is controlled only by the 'cost-out' configuration.

If you do a proc restart while the startup-cost-in timer is running, the node remains in cost-out state and the timer restarts.

### Configure EVPN Out of Service

This section describes how you can configure EVPN Out of Service.

```plaintext
/* Configuring node cost-out on a PE */
Router# configure
Router(config)# evpn
Router(config-evpn)# cost-out
Router(config-evpn) commit

/* Bringing up the node into service */
Router# configure
Router(config)# evpn
Router(config-evpn)# no cost-out
Router(config-evpn) commit

/* Configuring the timer to bring up the node into service after the specified time on reload */
Router# configure
Router(config)# evpn
Router(config-evpn)# startup-cost-in 6000
Router(config-evpn) commit
```

### Running Configuration

```plaintext
configure
evpn
  cost-out
!

configure
evpn
  startup-cost-in 6000
!
```

### Verification

Verify the EVPN Out of Service configuration.
/* Verify the node cost-out configuration */

Router# show evpn summary
Fri Apr 7 07:45:22.311 IST
Global Information
----------------------------------------
Number of EVIs : 2
Number of Local EAD Entries : 0
Number of Remote EAD Entries : 0
Number of Local MAC Routes : 5
    MAC : 5
    MAC-IPv4 : 0
    MAC-IPv6 : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes : 7
    MAC : 7
    MAC-IPv4 : 0
    MAC-IPv6 : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels : 5
Number of ES Entries : 9
Number of Neighbor Entries : 1
EVPN Router ID : 192.168.0.1
BGP Router ID : ::
BGP ASN : 100
PBB BSA MAC address : 0207.1fee.be00
Global peering timer : 3 seconds
Global recovery timer : 30 seconds
EVPN cost-out : TRUE
    startup-cost-in timer : Not configured

/* Verify the no cost-out configuration */

Router# show evpn summary
Fri Apr 7 07:45:22.311 IST
Global Information
----------------------------------------
Number of EVIs : 2
Number of Local EAD Entries : 0
Number of Remote EAD Entries : 0
Number of Local MAC Routes : 5
    MAC : 5
    MAC-IPv4 : 0
    MAC-IPv6 : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes : 7
    MAC : 7
    MAC-IPv4 : 0
    MAC-IPv6 : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels : 5
Number of ES Entries : 9
Number of Neighbor Entries : 1
EVPN Router ID : 192.168.0.1
BGP Router ID : ::
BGP ASN : 100
PBB BSA MAC address : 0207.1fee.be00
Global peering timer : 3 seconds
EVPN Multiple Services per Ethernet Segment

EVPN Multiple Services per Ethernet Segment feature allows you to configure multiple services over single Ethernet Segment (ES). Instead of configuring multiple services over multiple ES, you can configure multiple services over a single ES.

You can configure the following services on a single Ethernet Bundle; you can configure one service on each sub-interface.

- Flexible cross-connect (F XC) service. It supports VLAN Unaware, VLAN Aware, and Local Switching modes.

  For more information, see Configure Point-to-Point Layer 2 Services chapter in L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5500540 Series Routers.

- EVPN-VPWS Xconnect service

  For more information, see EVPN Virtual Private Wire Service (VPWS) chapter in L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5500540 Series Routers.

- EVPN Integrated Routing and Bridging (IRB)
Configure EVVPN Multiple Services per Ethernet Segment

Consider a customer edge (CE) device connected to two provider edge (PE) devices through Ethernet Bundle interface 22001. Configure multiple services on Bundle Ethernet sub-interfaces.

**Configuration Example**

Consider Bundle-Ether22001 ES, and configure multiple services on sub-interface.

```bash
/* Configure attachment circuits */
Router(config)# configure
Router(config)# interface Bundle-Ether22001.12 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 12
Router(config-l2vpn-subif)# exit
Router(config)# interface Bundle-Ether22001.13 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 13
Router(config-l2vpn-subif)# exit
Router(config)# interface Bundle-Ether22001.14 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 14
Router(config-l2vpn-subif)# exit
Router(config)# interface Bundle-Ether22001.1 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 1
Router(config-l2vpn-subif)# exit
Router(config)# interface Bundle-Ether22001.2 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 2
Router(config-l2vpn-subif)# exit
Router(config)# interface Bundle-Ether22001.3 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 3
Router(config-l2vpn-subif)# exit
Router(config)# interface Bundle-Ether22001.4 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 4
Router(config-l2vpn-subif)# exit
Router(config)# exit

/*Configure VLAN Unaware FXC Service */
Router(config)# l2vpn
Router(config-l2vpn)# flexible-xconnect-service vlan-unaware fxc_mh1
Router(config-l2vpn-fxs-vu)# interface Bundle-Ether22001.1
Router(config-l2vpn-fxs-vu)# interface Bundle-Ether22001.2
Router(config-l2vpn-fxs-vu)# interface Bundle-Ether22001.3
Router(config-l2vpn-fxs-vu)# neighbor evpn evi 21006 target 22016
Router(config-l2vpn-fxs-vu)# commit
```

All these services are supported only on all-active multihoming scenario.
/* Configure VLAN Aware FXC Service */
Router(config)# 12vpn
Router(config-12vpn)# flexible-xconnect-service vlan-aware evi 24001
Router(config-12vpn-fxs-va)# interface Bundle-Ether22001.12
Router(config-12vpn-fxs-va)# interface Bundle-Ether2201.13
Router(config-12vpn-fxs-va)# interface Bundle-Ether22001.14
Router(config-12vpn-fxs-va)# commit

/* Configure Local Switching - Local switching is supported only on VLAN-aware FXC */
PE1
Router# configure
Router(config)# 12vpn
Router(config-12vpn)# flexible-xconnect-service vlan-aware evi 31400
Router(config-12vpn-fxs-va)# interface Bundle-Ether22001.1400
Router(config-12vpn-fxs-va)# interface Bundle-Ether23001.1400
Router(config-12vpn-fxs-va)# commit
Router(config-12vpn-fxs-va)# exit
PE2
Router# configure
Router(config)# 12vpn
Router(config-12vpn)# flexible-xconnect-service vlan-aware evi 31401
Router(config-12vpn-fxs-va)# interface Bundle-Ether22001.1401
Router(config-12vpn-fxs-va)# interface Bundle-Ether23001.1401
Router(config-12vpn-fxs-va)# commit
Router(config-12vpn-fxs-va)# exit

/* Configure EVPN-VPWS xconnect service and native EVPN with IRB */

Router# configure
Router(config)# interface Bundle-Ether22001.11 l2transport
Router(config-12vpn-subif)# encapsulation dot1q 1 second-dot1q 11
Router(config-12vpn-subif)# rewrite ingress tag pop 2 symmetric
Router(config-12vpn-subif)# commit
Router(config-12vpn-subif)# exit

Router# configure
Router(config)# interface Bundle-Ether22001.21 l2transport
Router(config-12vpn-subif)# encapsulation dot1q 1 second-dot1q 21
Router(config-12vpn-subif)# rewrite ingress tag pop 2 symmetric
Router(config-12vpn-subif)# commit
Router(config-12vpn-subif)# exit

Router# configure
Router(config)# xconnect group xg22001
Router(config-12vpn-xc)# p2p evpn-vpws-mclag-22001
Router(config-12vpn-xc-p2p)# interface Bundle-Ether22001.11
Router(config-12vpn-xc-p2p)# neighbor evpn evi 22101 target 220101 source 220301
Router(config-12vpn-xc-p2p-pw)# commit
Router(config-12vpn-xc-p2p-pw)# exit

Router# configure
Router(config)# 12vpn
Router(config-12vpn)# bridge group native_evpn1
Router(config-12vpn-bg)# bridge-domain bd21
Router(config-12vpn-bg-bd)# interface Bundle-Ether22001.21
Router(config-12vpn-bg-bd-ac)# routed interface BVI21
Router(config-12vpn-bg-bd-bvi)# evi 22021
Router(config-12vpn-bg-bd-bvi)# commit
Router(config-12vpn-bg-bd-bvi)# exit

/* Configure Native EVPN */
Router (config-evpn)# evpn
Router (config-evpn)# interface Bundle-Ether22001
Router (config-evpn-ac)# ethernet-segment identifier type 0 ff ff ff ff ff ff ff ff ee
Router (config-evpn-ac-es)# bgp route-target 2200.0001.0001
Router (config-evpn-ac-es)# exit
Router (config-evpn)# evi 24001
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64:24001
Router (config-evpn-evi-bgp)# route-target export 64:24001
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# evi 21006
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target route-target 64:10000
Router (config-evpn-evi-bgp)# exit
Router (config-evpn)# evi 22101
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64:22101
Router (config-evpn-evi-bgp)# route-target export 64:22101
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# exit
Router (config-evpn)# evi 22021
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64:22021
Router (config-evpn-evi-bgp)# route-target export 64:22021
Router (config-evpn-evi-bgp)# exit
Router (config-evpn)# evi 22022
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64:22022
Router (config-evpn-evi-bgp)# route-target export 64:22022
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# advertise-mac
Router (config-evpn-evi)# commit
Router (config-evpn-evi)# exit

Running Configuration

/* Configure attachment circuits */
interface Bundle-Ether22001.12 l2transport
encapsulation dot1q 1 second-dot1q 12
!
interface Bundle-Ether22001.13 l2transport
encapsulation dot1q 1 second-dot1q 13
!
interface Bundle-Ether22001.14 l2transport
encapsulation dot1q 1 second-dot1q 14
!
interface Bundle-Ether22001.1 l2transport
encapsulation dot1q 1 second-dot1q 1
!
interface Bundle-Ether22001.2 l2transport
encapsulation dot1q 1 second-dot1q 2
!
interface Bundle-Ether22001.3 l2transport
encapsulation dot1q 1 second-dot1q 3
! interface Bundle-Ether22001.4 l2transport
capsulation dot1q 1 second-dot1q 4

/*Configure VLAN Unaware FXC Service */
flexible-xconnect-service vlan-unaware fxc_mh1
   interface Bundle-Ether22001.1
   interface Bundle-Ether22001.2
   interface Bundle-Ether22001.3
neighbor evpn evi 21006 target 22016
!
/*Configure VLAN Aware FXC Service */
l2vpn
   flexible-xconnect-service vlan-aware evi 24001
      interface Bundle-Ether22001.12
      interface Bundle-Ether22001.13
      interface Bundle-Ether22001.14

/* Configure Local Switching */
flexible-xconnect-service vlan-aware evi 31400
   interface Bundle-Ether22001.1400
   interface Bundle-Ether23001.1400
!
flexible-xconnect-service vlan-aware evi 31401
   interface Bundle-Ether22001.1401
   interface Bundle-Ether23001.1401
!
/* Configure EVPN-VPWS xconnect service and native EVPN with IRB */
interface Bundle-Ether22001.11 l2transport
   encapsulation dot1q 1 second-dot1q 11
   rewrite ingress tag pop 2 symmetric
!
interface Bundle-Ether22001.21 l2transport
   encapsulation dot1q 1 second-dot1q 21
   rewrite ingress tag pop 2 symmetric
!
!
l2vpn
xconnect group xg22001
p2p evpn-vpws-mclag-22001
   interface Bundle-Ether22001.11
neighbor evpn evi 22101 target 220101 source 220301
!
bridge group native_evpn1
   bridge-domain bd21
      interface Bundle-Ether22001.21
      routed interface BVI21
         evi 22021
!
/* Configure Native EVPN */
Evpn
   interface Bundle-Ether22001
      ethernet-segment identifier type 0 ff.ff.ff.ff.ff.ff.ee
      bgp route-target 2200.0001.0001
!
evi 24001
   bgp
      route-target import 64:24001
      route-target export 64:24001
!
evi 21006
   bgp
      route-target 64:100006
Verification

Verify if each of the services is configured on the sub-interface.

Router# show l2vpn xconnect summary
Number of groups: 6
Number of xconnects: 505 Up: 505 Down: 0 Unresolved: 0 Partially-programmed: 0
AC-PW: 505 AC-AC: 0 PW-PW: 0 Monitor-Session-PW: 0
Number of Admin Down segments: 0
Number of MP2MP xconnects: 0
Up 0 Down 0
Advertised: 0 Non-Advertised: 0

Router# show l2vpn flexible-xconnect-service summary
Number of flexible xconnect services: 74
Up: 74

Router# show l2vpn flexible-xconnect-service name fxc mh1
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (FP) = Partially Programmed
Flexible XConnect Service Segment
Name  ST  Type  Description  ST
---------------------------------- ----------------------------------
fxc mh1  UP  AC:  BE22001.1  UP
        AC:  BE22001.2  UP
        AC:  BE22001.3  UP
----------------------------------

Router# show l2vpn flexible-xconnect-service name evi:24001
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (FP) = Partially Programmed
Flexible XConnect Service Segment
Name  ST  Type  Description  ST
------------------- ----------------- ----------------------------------
evi:24001  UP  AC:  BE22001.11  UP
        AC:  BE22001.12  UP
        AC:  BE22001.13  UP
Router# show l2vpn xconnect group xg22001 xc-name evpn-vpws-mclag-22001
Fri Sep 1 17:28:58.259 UTC
XConnect Segment 1 Segment 2
Group Name ST Description ST Description ST
------------------------ ----------------------------- -----------------------------------
xg22001 evpn-vpws-mclag-22001 UP BE22001.101 UP EVPN 22101, 220101,64.1.1.6 UP
------------------------------------------------------------------------------------------

Associated Commands

- evpn
- evi
- ethernet-segment
- advertise-mac
- show evpn ethernet-segment
- show evpn evi
- show evpn summary
- show l2vpn xconnect summary
- show l2vpn flexible-xconnect-service
- show l2vpn xconnect group

EVPN Routing Policy

The EVPN Routing Policy feature provides the route policy support for address-family L2VPN EVPN. This feature adds EVPN route filtering capabilities to the routing policy language (RPL). The filtering is based on various EVPN attributes.

A routing policy instructs the router to inspect routes, filter them, and potentially modify their attributes as they are accepted from a peer, advertised to a peer, or redistributed from one routing protocol to another.

This feature enables you to configure route-policies using EVPN network layer reachability information (NLRI) attributes of EVPN route type 1 to 5 in the route-policy match criteria, which provides more granular definition of route-policy. For example, you can specify a route-policy to be applied to only certain EVPN route-types or any combination of EVPN NLRI attributes. This feature provides flexibility in configuring and deploying solutions by enabling route-policy to filter on EVPN NLRI attributes.

To implement this feature, you need to understand the following concepts:

- Routing Policy Language
- Routing Policy Language Structure
- Routing Policy Language Components
• Routing Policy Language Usage
• Policy Definitions
• Parameterization
• Semantics of Policy Application
• Policy Statements
• Attach Points

For information on these concepts, see Implementing Routing Policy.

Currently, this feature is supported only on BGP neighbor “in” and "out" attach points. The route policy can be applied only on inbound or outbound on a BGP neighbor.

**EVPN Route Types**

The EVPN NLRI has the following different route types:

**Route Type 1: Ethernet Auto-Discovery (AD) Route**

The Ethernet (AD) routes are advertised on per EVI and per Ethernet Segment Identifier (ESI) basis. These routes are sent per Ethernet segment (ES). They carry the list of EVIs that belong to the ES. The ESI field is set to zero when a CE is single-homed.

An Ethernet A-D route type specific EVPN NLRI consists of the following fields:

```
+-------------------------------+   +-----------+           +-------------------+   +-----------+
| Route Type (1 octet)               |   | Length (1 octet)                          |   | Route Distinguish (RD) (8 octets)   |
+-------------------------------+   +-----------+           +-------------------+   +-----------+
| Route Distinguish (RD) (8 octets)   |   | Ethernet Segment Identifier (10 octets) |
+-------------------------------+   +-----------+           +-------------------+   +-----------+
| Ethernet Segment Identifier (10 octets) |
+-------------------------------+   +-----------+           +-------------------+   +-----------+
| Ethernet Tag ID (4 octets)           |   | MPLS Label (3 octets)                     |
+-------------------------------+   +-----------+           +-------------------+   +-----------+
| MPLS Label (3 octets)               |
```

**NLRI Format: Route-type 1:**

```
[Type][Len][RD][ESI][ETag][MPLS Label]
```

**Net attributes:** [Type][RD][ESI][ETag]

**Path attributes:** [MPLS Label]

**Example**

```
route-policy evpn-policy
  if rd in (1.1.1.1:0) [and/or evpn-route-type is 1] [and/or esi in (0a1.a2a3.a4a5.a6a7.a8a9)]
  (and/or etag is 4294967295) then
    set ..
```
Route Type 2: MAC/IP Advertisement Route

The host's IP and MAC addresses are advertised to the peers within NLRI. The control plane learning of MAC addresses reduces unknown unicast flooding.

A MAC/IP Advertisement Route type specific EVPN NLRI consists of the following fields:

```
+----------------------------------+
| Route Type (1 octet)             |
+----------------------------------+
| Length (1 octet)                 |
+----------------------------------+
| RD (8 octets)                   |
+----------------------------------+
| Ethernet Segment Identifier (10 octets) |
+----------------------------------+
| Ethernet Tag ID (4 octets)       |
+----------------------------------+
| MAC Address Length (1 octet)     |
+----------------------------------+
| MAC Address (6 octets)           |
+----------------------------------+
| IP Address Length (1 octet)      |
+----------------------------------+
| IP Address (0, 4, or 16 octets)  |
+----------------------------------+
| MPLS Label1 (3 octets)           |
+----------------------------------+
| MPLS Label2 (0 or 3 octets)      |
+----------------------------------+
```

NLRI Format: Route-type 2:

```
[Type][Len][RD][ESI][ETag][MAC Addr Len][MAC Addr][IP Addr Len][IP Addr][MPLS Label1][MPLS Label2]
```

Net attributes: [Type][RD][ETag][MAC Addr Len][MAC Addr][IP Addr Len][IP Addr]

Path attributes: [ESI], [MPLS Label1], [MPLS Label2]
Example

route-policy evpn-policy
    if rd in (1.1.1.2:0) [and/or evpn-route-type is 2] [and/or esi in
    (0000.0000.0000.0000.0000) [and/or etag is 0] [and/or macaddress in (0013.aabb.ccdd)]
    [and/or destination in (1.2.3.4/32)] then
        set ..
    endif
end-policy

Route Type 3: Inclusive Multicast Ethernet Tag Route

This route establishes the connection for broadcast, unknown unicast, and multicast (BUM) traffic from asource PE to a remote PE. This route is advertised on per VLAN and per ESI basis.

An Inclusive Multicast Ethernet Tag route type specific EVPN NLRI consists of the following fields:

+------------------------------------+  
| Route Type (1 octet)            | *  
+------------------------------------+  
| Length (1 octet)                |  
+------------------------------------+  
| RD (8 octets)                   | *  
+------------------------------------+  
| Ethernet Tag ID (4 octets)      | *  
+------------------------------------+  
| IP Address Length (1 octet)     | *  
+------------------------------------+  
| Originating Router's IP Address | *  
| (4 or 16 octets)                |  
+------------------------------------+  

NLRI Format: Route-type 3:

[Type][Len][RD][ETag][IP Addr Len][Originating Router's IP Addr]

Net attributes: [Type][RD][ETag][IP Addr Len][Originating Router's IP Addr]

Example

route-policy evpn-policy
    if rd in (1.1.1.1:300) [and/or evpn-route-type is 3] [and/or etag is 0] [and/or
evpn-originator in (1.1.1.1)] then
        set ..
    endif
end-policy
Route Type 4: Ethernet Segment Route

Ethernet segment routes enable to connect a CE device to two or PE devices. ES route enables the discovery of connected PE devices that are connected to the same Ethernet segment.

An Ethernet Segment route type specific EVPN NLRI consists of the following fields:

```
+-----------------------------------------------------+
| Route Type (1 octet)                                |
+-----------------------------------------------------+
| Length (1 octet)                                    |
+-----------------------------------------------------+
| RD (8 octets)                                        |
+-----------------------------------------------------+
| Ethernet Segment Identifier (10 octets)             |
+-----------------------------------------------------+
| IP Address Length (1 octet)                         |
+-----------------------------------------------------+
| Originating Router's IP Address                     |
| (4 or 16 octets)                                    |
+-----------------------------------------------------+
```

**NLRI Format: Route-type 4:**

```
[Type][Len][RD][ESI][IP Addr Len][Originating Router's IP Addr]
```

**Net attributes:**

```
[Type][RD][ESI][IP Addr Len][Originating Router's IP Addr]
```

**Example**

```
route-policy evpn-policy
  if rd in (1.1.1.1:0) [and/or evpn-route-type is 4] [and/or esi in (00a1.a2a3.a4a5.a6a7.a8a9)] [and/or evpn-originator in (1.1.1.1)] then
  set ..
endif
end-policy
```

Route Type 5: IP Prefix Route

An IP Prefix Route type specific EVPN NLRI consists of the following fields:
EVPN Features

EVPN RPL Attribute

Route Distinguisher
A Route Distinguisher (rd) attribute consists of eight octets. An rd can be specified for each of the EVPN route types. This attribute is not mandatory in route-policy.

Example

```
rd in (1.2.3.4:0)
```

EVPN Route Type
EVPN route type attribute consists of one octet. This specifies the EVPN route type. The EVPN route type attribute is used to identify a specific EVPN NLRI prefix format. It is a net attribute in all EVPN route types.

Example

```
if rd in (30.30.30.30:1) [and/or evpn-route-type is 5] [and/or esi in (0000.0000.0000.0000.0000) [and/or etag is 0] [and/or destination in (12.2.0.0/16)] [and/or evpn-gateway in (0.0.0.0)] then
  set ..
endif
end-policy
```
Example

evpn-route-type is 3

The following are the various EVPN route types that can be used:
1 - ethernet-ad
2 - mac-advertisement
3 - inclusive-multicast
4 - ethernet-segment
5 - ip-advertisement

IP Prefix

An IP prefix attribute holds IPv4 or IPv6 prefix match specification, each of which has four parts: an address, a mask length, a minimum matching length, and a maximum matching length. The address is required, but the other three parts are optional. When IP prefix is specified in EVPN route type 2, it represents either a IPv4 or IPv6 host IP Address (/32 or /128). When IP prefix is specified in EVPN route type 5, it represents either IPv4 or IPv6 subnet. It is a net attribute in EVPN route type 2 and 5.

Example

destination in (128.47.10.2/32)
destination in (128.47.0.0/16)
destination in (128:47::1/128)
destination in (128:47::0/112)

esi

An Ethernet Segment Identifier (ESI) attribute consists of 10 octets. It is a net attribute in EVPN route type 1 and 4, and a path attribute in EVPN route type 2 and 5.

Example

esi in (ffff.ffff.ffff.ffff.fff0)

etag

An Ethernet tag attribute consists of four octets. An Ethernet tag identifies a particular broadcast domain, for example, a VLAN. An EVPN instance consists of one or more broadcast domains. It is a net attribute in EVPN route type 1, 2, 3 and 5.

Example

etag in (10000)
mac
The mac attribute consists of six octets. This attribute is a net attribute in EVPN route type 2.

Example

mac in (0206.acb1.e806)

evpn-originator
The evpn-originator attribute specifies the originating router's IP address (4 or 16 octets). This is a net attribute in EVPN route type 3 and 4.

Example

evpn-originator in (1.2.3.4)

evpn-gateway
The evpn-gateway attribute specifies the gateway IP address. The gateway IP address is a 32-bit or 128-bit field (IPv4 or IPv6), and encodes an overlay next-hop for the IP prefixes. The gateway IP address field can be zero if it is not used as an overlay next-hop. This is a path attribute in EVPN route type 5.

Example

evpn-gateway in (1.2.3.4)

EVPN RPL Attribute Set
In this context, the term set is used in its mathematical sense to mean an unordered collection of unique elements. The policy language provides sets as a container for groups of values for matching purposes. Sets are used in conditional expressions. The elements of the set are separated by commas. Null (empty) sets are allowed.

prefix-set
A prefix-set holds IPv4 or IPv6 prefix match specifications, each of which has four parts: an address, a mask length, a minimum matching length, and a maximum matching length. The address is required, but the other three parts are optional. The prefix-set specifies one or more IP prefixes.

Example

prefix-set ip_prefix_set
14.2.0.0/16,
54.0.0.0/16,
12.12.12.0/24,
50:50::1:0/112
end-set
**mac-set**

The mac-set specifies one or more MAC addresses.

**Example**

```plaintext
mac-set mac_address_set
1234.2345.6789,
2345.3456.7890
end-set
```

**esi-set**

The esi-set specifies one or more ESI's.

**Example**

```plaintext
esi-set evpn_esi_set
1234.2345.3456.4567.5678,
1234.2345.3456.4567.5670
end-set
```

**etag-set**

The etag-set specifies one or more Ethernet tags.

**Example**

```plaintext
etag-set evpn_etag_set
10000,
20000
end-set
```

---

**Configure EVPN RPL Feature**

The following section describe how to configure mac-set, esi-set, evpn-gateway, and evpn-originator.

```plaintext
/* Configuring a mac-set and refering it in a route-policy (Attach point = neighbor-in) */
Router# configure
Router(config)# mac-set demo_mac_set
Router(config-mac)# 1234.ffff.aaa3,
Router(config-mac)# 2323.4444.ffff
Router(config-mac)# end-set
Router(config)# route-policy policy_use_pass_mac_set
Router(config-rpl)# if mac in demo_mac_set then
  Router(config-rpl-if)# set med 200
  Router(config-rpl-if)# else
  Router(config-rpl-else)# set med 1000
  Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Router(config)# commit
```
Router(config)# router bgp 100
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# !
Router(config-bgp-af)# neighbor 10.0.0.10
Router(config-bgp-nbr)# remote-as 8
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy policy_use_pass_mac_set in
Router(config-bgp-nbr-af)# commit

/* Configuring a esi-set and referring it in a route-policy (Attach point - neighbor-in) */
Router# configure
Router(config)# esi-set demo_esi
Router(config-esi)# ad34.1233.1222.ffff.44ff,
Router(config-esi)# ad34.1233.1222.ffff.6666
Router(config-esi)# end-set
Router(config)#
Router(config)# route-policy use_esi
Router(config-rpl)# if esi in demo_esi then
Router(config-rpl-if)# set local-preference 100
Router(config-rpl-if)# else
Router(config-rpl-else)# set local-preference 300
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Router(config)# commit

/* Configuring evpn-gateway/evpn-originator in a route-policy (Attach point - neighbor-in and out) */
Router# configure
Router(config)# route-policy gateway_demo
Router(config-rpl)# if evpn-gateway in (10.0.0.0/32) then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
Router(config-rpl)# end-policy
Router(config)# commit
Router(config)# route-policy originator_demo
Router(config-rpl)# if evpn-originator in (10.0.0.1/32) then
Router(config-rpl-if)# set local-preference 100
Router(config-rpl-if)# else
Router(config-rpl-else)# set med 200
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Router(config)# commit
Router(config)# router bgp 100
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# !
Router(config-bgp-af)# neighbor 10.0.0.10
Router(config-bgp-nbr)# remote-as 8
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy gateway_demo in
Router(config-bgp-nbr-af)# route-policy originator_demo out
Router(config-bgp-nbr-af)# commit

Running Configuration

/* Configuring a mac-set and referring it in a route-policy (Attach point - neighbor-in) */
mac-set demo_mac_set
  1234.ffff.aaa3,
  2323.4444.ffff
end-set
route-policy policy_use_pass_mac_set
if mac in demo_mac_set then
    set med 200
else
    set med 1000
endif
eンド-policy!

router bgp 100
  address-family ipv4 unicast
  !
  neighbor 10.0.0.10
    remote-as 8
    address-family ipv4 unicast
    route-policy policy_use_pass_mac_set in
  !
  !
end

/* Configuring a esi-set and refering it in a route-policy (Attach point - neighbor-in) */
Wed Oct 26 11:52:23.720 IST
esi-set demo_esi
  ad34.1233.1222.ffff.44ff,
  ad34.1233.1222.ffff.6666
end-set!

route-policy use_esi
  if esi in demo_esi then
    set local-preference 100
  else
    set local-preference 300
  endif
eンド-policy

**EVPN Route Policy Examples**

route-policy ex_2
  if rd in (2.2.18.2:1004) and evpn-route-type is 1 then
    drop
  elseif rd in (2.2.18.2:1009) and evpn-route-type is 1 then
    drop
  else
    pass
  endif
eンド-policy!

route-policy ex_3
  if evpn-route-type is 5 then
    set extcommunity bandwidth (100:9999)
  else
    pass
  endif
eンド-policy!

route-policy samp
eンド-policy!

route-policy samp1
  if rd in (30.0.101.2:0) then
    pass
  endif
eンド-policy
route-policy samp2
   if rd in (30.0.101.2:0, 1:1) then
       pass
   endif
end-policy

route-policy samp3
   if rd in ("*:*") then
       pass
   endif
end-policy

route-policy samp4
   if rd in (30.0.101.2:*) then
       pass
   endif
end-policy

route-policy samp5
   if evpn-route-type is 1 then
       pass
   endif
end-policy

route-policy samp6
   if evpn-route-type is 2 or evpn-route-type is 5 then
       pass
   endif
end-policy

route-policy samp7
   if evpn-route-type is 4 or evpn-route-type is 3 then
       pass
   endif
end-policy

route-policy samp8
   if evpn-route-type is 1 or evpn-route-type is 2 or evpn-route-type is 3 then
       pass
   endif
end-policy

route-policy samp9
   if evpn-route-type is 1 or evpn-route-type is 2 or evpn-route-type is 3 or evpn-route-type is 4 then
       pass
   endif
end-policy

route-policy test1
   if evpn-route-type is 2 then
       set next-hop 10.2.3.4
   else
       pass
   endif
end-policy

route-policy test2
   if evpn-route-type is 2 then
       set next-hop 10.10.10.10
   else
       drop
   endif
end-policy
!
route-policy test3
  if evpn-route-type is 1 then
    set tag 9988
  else
    pass
  endif
end-policy!
route-policy samp21
  if mac in (6000.6000.6000) then
    pass
  endif
end-policy!
route-policy samp22
  if extcommunity rt matches-any (100:1001) then
    pass
  else
    drop
  endif
end-policy!
route-policy samp23
  if evpn-route-type is 1 and esi in (aaaa.bbbb.cccc.dddd.eeee) then
    pass
  else
    drop
  endif
end-policy!
route-policy samp24
  if evpn-route-type is 5 and extcommunity rt matches-any (100:1001) then
    pass
  else
    drop
  endif
end-policy!
route-policy samp25
  if evpn-route-type is 2 and esi in (1234.1234.1234.1234.1236) then
    pass
  else
    drop
  endif
end-policy!
route-policy samp26
  if etag in (20000) then
    pass
  else
    drop
  endif
end-policy!
route-policy samp27
  if destination in (99.99.99.1) and etag in (20000) then
    pass
  else
    drop
  endif
end-policy!
route-policy samp31
  if evpn-route-type is 1 or evpn-route-type is 2 or evpn-route-type is 3 or evpn-route-type is 4 or evpn-route-type is 5 then
    pass
  else
    drop
  endif
end-policy

route-policy samp33
  if esi in evpn_esi_set1 then
    pass
  else
    drop
  endif
end-policy

route-policy samp34
  if destination in (90:1:1::9/128) then
    pass
  else
    drop
  endif
end-policy

route-policy samp35
  if destination in evpn_prefix_set1 then
    pass
  else
    drop
  endif
end-policy

route-policy samp36
  if evpn-route-type is 3 and evpn-originator in (80:1:1::3) then
    pass
  else
    drop
  endif
end-policy

route-policy samp37
  if evpn-gateway in (10:10::10) then
    pass
  else
    drop
  endif
end-policy

route-policy samp38
  if mac in evpn_mac_set1 then
    pass
  else
    drop
  endif
end-policy

route-policy samp39
  if mac in (6000.6000.6002) then
    pass
  else
    drop
  endif
end-policy
route-policy samp41
  if evpn-gateway in (10.10.10.10, 10:10::10) then
    pass
  else
    drop
  endif
end-policy

route-policy samp42
  if evpn-originator in (24.162.160.1/32, 70:1:1::1/128) then
    pass
  else
    drop
  endif
end-policy

route-policy example
  if rd in (62300:1903) and evpn-route-type is 1 then
    drop
  elseif rd in (62300:19032) and evpn-route-type is 1 then
    drop
  else
    pass
  endif
end-policy

route-policy samp100
  if evpn-route-type is 4 or evpn-route-type is 5 then
    drop
  else
    pass
  endif
end-policy

route-policy samp101
  if evpn-route-type is 4 then
    drop
  else
    pass
  endif
end-policy

route-policy samp102
  if evpn-route-type is 4 then
    drop
  elseif evpn-route-type is 5 then
    drop
  else
    pass
  endif
end-policy

route-policy samp103
  if evpn-route-type is 2 and destination in evpn_prefix_set1 then
    drop
  else
    pass
  endif
end-policy

route-policy samp104
  if evpn-route-type is 1 and etag in evpn_etag_set1 then
    drop
elseif evpn-route-type is 2 and mac in evpn_mac_set1 then
drop
elseif evpn-route-type is 5 and esi in evpn_esi_set1 then
drop
else
    pass
endif
end-policy