EVPN Features

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

- EVPN Overview, on page 2
- EVPN Concepts, on page 3
- EVPN Operation, on page 4
- EVPN Route Types, on page 5
- EVPN Timers, on page 6
- Configure EVPN L2 Bridging Service, on page 7
- EVPN Single-Active Multi-Homing Mode, on page 9
- EVPN Software MAC Learning , on page 15
- EVPN Out of Service, on page 23
- CFM Support for EVPN, on page 26
- Control Word Support for ELAN, on page 27
- EVPN Multiple Services per Ethernet Segment, on page 27
- EVPN Single-Flow-Active Multihoming Load-Balancing Mode, on page 33
- EVPN Convergence Using NTP Synchronization, on page 39
- **EVPN MPLS Seamless Integration with VPLS** , on page 41
- Configure EVPN on the Existing VPLS Network, on page 42
- EVI Configuration Under L2VPN Bridge-Domain, on page 45
- Verify EVPN Configuration, on page 46
- Clear Forwarding Table, on page 49
- Hierarchical EVPN Access Pseudowire, on page 49
- Network Convergence using Core Isolation Protection, on page 51
- Configurable Recovery Time for EVPN Core Isolation Group, on page 58
- Conditional Advertisement of Default-Originate, on page 65
- EVPN Core Isolation Protection, on page 68
- EVPN Routing Policy, on page 70
- CFM on EVPN ELAN, on page 86
- EVPN Bridging and VPWS Services over BGP-LU Underlay, on page 94
- Support for DHCPv4 and DHCPv6 Client over BVI, on page 106

EVPN Overview

Ethernet VPN (EVPN) is a 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 multihoming 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 multihoming 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 - Enables you to connect a customer edge (CE) device to one provider edge (PE) device.
- Multihoming - 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 Restrictions

When paths of different technologies are resolved over ECMP, it results in *heterogeneous* ECMP, leading to severe network traffic issues. Don't use ECMP for any combination of the following technologies:

- LDP.
- BGP-LU, including services over BGP-LU loopback peering or recursive services at Level-3
- VPNv4.
- 6PE and 6VPE.
- EVPN.
- Recursive static routing.

EVPN Concepts

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

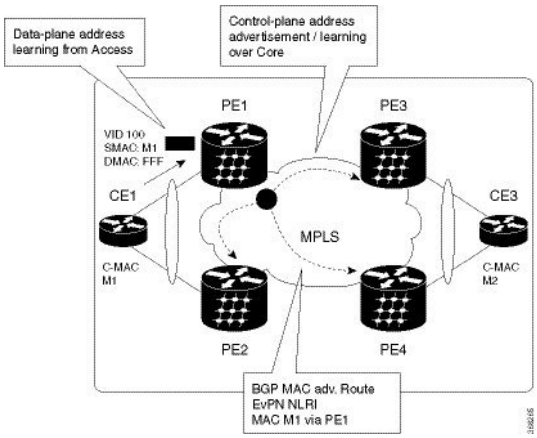
- Ethernet Segment (ES): An Ethernet segment is a set of Ethernet links that connects a multihomed device. If a multi-homed device or network is connected to two or more PEs through a set of Ethernet links, then that set of links is referred to as an Ethernet segment. The Ethernet segment route is also 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



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.

Behavior Change due to ESI Label Assignment

To adhere to RFC 7432 recommendations, the encoding or decoding of MPLS label is modified for extended community. Earlier, the lower 20 bits of extended community were used to encode the split-horizon group (SHG) label. Now, the SHG label encoding uses from higher 20 bits of extended community.

According to this change, routers in same ethernet-segment running old and new software release versions decodes extended community differently. This change causes inconsistent SHG labels on peering EVPN PE routers. Almost always, the router drops BUM packets with incorrect SHG label. However, in certain conditions, it may cause remote PE to accept such packets and forward to CE potentially causing a loop. One such instance is when label incorrectly read as NULL.

To overcome this problem, Cisco recommends you to:

- Minimize the time both PEs are running different software release versions.
- Before upgrading to a new release, isolate the upgraded node and shutdown the corresponding AC bundle.
- After upgrading both the PEs to the same release, you can bring both into service.

Similar recommendations are applicable to peering PEs with different vendors with SHG label assignment that does not adhere to RFC 7432.

EVPN Route Types

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

Table 1: EVPN Route Types

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

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.

EVPN Timers

The following table shows various EVPN timers:

Table 2: EVPN Timers

Timer	Range	Default Value	Trigger	Applicability	Action	Sequence
global mac evpn timer	0-300s	300s	when BGP is fired	Single-Flow-Active and Multi homed all active	Delay the time and effort required to delete the remote portion to save programming cycles working for forwarding path first.	4

Global MAC EVPN Timer

Global mac evpn timer is configurable under **evpn timers mac-postpone** timer. Global MAC EVPN timer is relevant for SYNC routes only in the following scenarios:

- FRR (fast re-route)is configured: MAC and MAC+IP deletes are postponed to help with convergence.

- All-active: MAC+IPs deletes are postponed to allow time for ARP to converge.
- Single-flow-active: MAC+IP deletes are postponed to allow speculative (Address Resolution Protocol) ARP to point to local adjacency.

Typically, a route that is deleted is always quickly learned locally. Using this knowledge, we can delay the time and effort required to delete the remote portion to save programming cycles working for forwarding path first.

Note The timer of 5-minutes start when EVPN receives a delete from BGP. The timer doesn't start at the exact time of AC shut or mass-withdraw.

The benefit of this speculative behavior is that we can reduce MAC-IP delete/re-create churn in forwarding and BGP.

Triggers of Global Mac EVPN Timer:

- Shut / No shut on IRB/BVI Interfaces.
- Removing and adding AC Interface Configuration.
- Removing and adding BVI Interface Configuration.
- Removing and adding BVI Interface from Bridge Domains.
- Shut / No shut on AC/Main-port Interface Configuration.

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 A device can contain up to 128K MAC address entries. A bridge domain on a device can contain up to 64K MAC address entries.

Note Flooding disable isn’t supported on EVPN bridge domains.

```
/* 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-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

/* Configure EVI and define the corresponding BGP route targets */
```

Note EVI route target used for multicast EVPN supports only extcomm type sub-type 0xA for EVI route target, the two-octet Autonomous System (AS) specific Extended Community. This means that when using a 4-byte AS number for BGP, you must additionally configure BGP import and export route targets under the EVPN configuration.

```
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
Router(config-l2vpn-bg)# bridge-domain 1-1
Router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/1
Router(config-l2vpn-bg-bd-ac)# evi 6005
Router(config-l2vpn-bg-bd-ac-evi)# commit
Router(config-l2vpnbg-bd-ac-evi)# exit
```

Running Configuration

```
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
  updatesource Loopback0
  addressfamily l2vpn evpn
!

configure
evpn
 evi 6005
  bgp
   rd 200:50
   route-target import 100:6005
   route-target export 100:6005
!
  advertise-mac
```

```
configure
l2vpn
 bridge group 1
  bridge-domain 1-1
   interface GigabitEthernet 0/0/0/1

    evi 6005
!
```

EVPN Single-Active Multi-Homing Mode

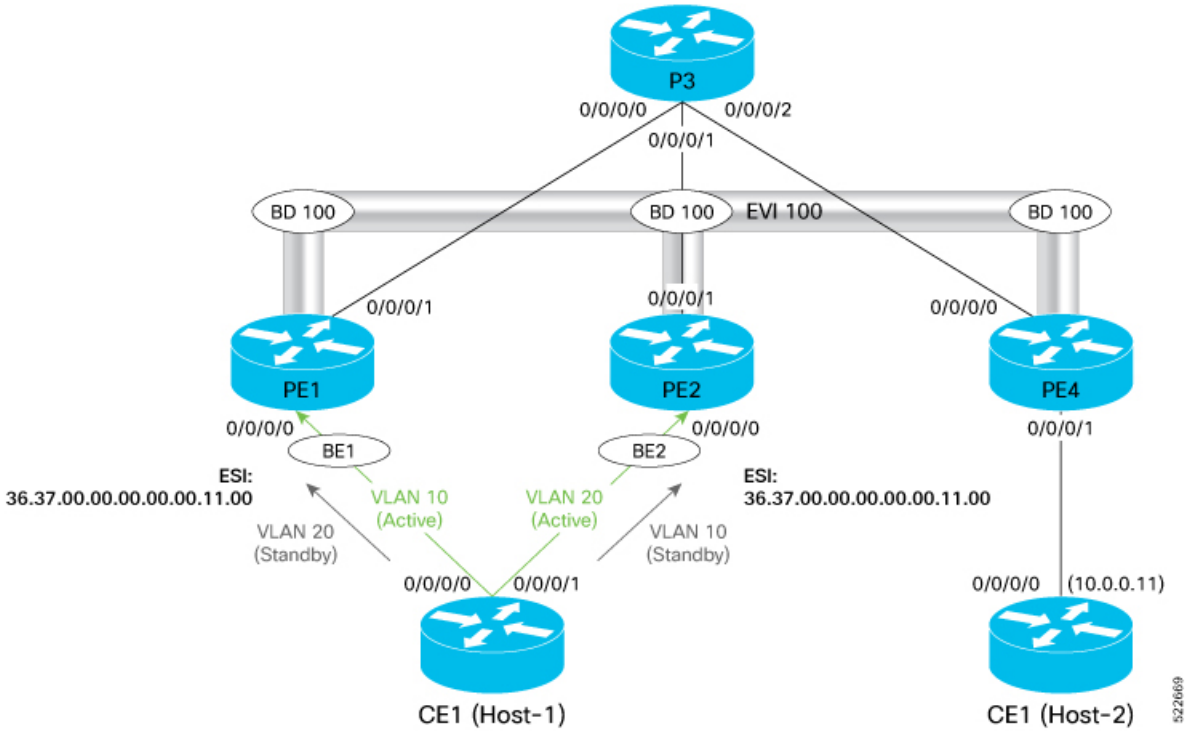
In single-active multihoming mode, only a single edge (PE) Router among a group of PE Routers attached to a host is allowed to send and recieve traffic on a given VLAN.

The single-active mode offers redundant connectivity for a VLAN on a single link at a time with failover to the second link in case the active link fails. The single-active mode directs the traffic to a single uplink. This mode is useful for network scenarios where policing, metering, and billing are required.

In Single-Active mode, Cisco IOS XR sends a topology change notification on the Ethernet segment links when a service carving update occurs, so that CEs flush their MAC tables and redirect traffic to the new DF-Elected PE.

Topology

Let's understand how the single-active mode works with this sample topology.



In this topology,

- The CE Router is multihomed to PE1 and PE2. Only one active uplink is allowed to send and receive traffic at any given time.

- In this mode, each link towards PE is in a unique ethernet bundle interface. In this example, BE1 is the ethernet bundle interface connecting CE1 and PE1. BE2 is the ethernet bundle interface connecting CE1 and PE2.
- As both the links are in a separate ethernet bundle interface, CE1 floods traffic at first to both the PE devices, but only the PE that is the Designated Forwarder (DF) forwards the traffic.
- In this mode, the uplinks to PE1 and PE2 are individual links and by default, the host chooses the DF uplink for forwarding for a given VLAN.

Configure EVPN Single-Active Multi-Homing

Perform the following tasks to configure EVPN single-active multi-homing:

- Configure Ethernet bundles on CE1 for multi-homing.
- Configure EVPN based single-active multi-homing.

Note Sub-interface shutdown is not supported in Single-Active load-balancing mode

Configure Ethernet bundles on CE1 for Multihoming:

```
Router#configure
Router(config)#interface Bundle-Ether1
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Bundle-Ether2
Router(config-if)# no shutdown
Router(config)#exit
Router(config)#interface HundredGigE0/0/0/0
Router(config-if)#bundle id 1 mode active
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface HundredGigE0/0/0/1
Router(config-if)#bundle id 2 mode active
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface HundredGigE0/0/0/2
Router(config-if)#exit
Router(config)#interface HundredGigE0/0/0/3
Router(config-if)#no shutdown
Router(config-if)#commit
Router(config-if)#exit
Router(config)#interface Bundle-Ether1.10 l2transport
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#rewrite ingress tag pop 1 symmetric
Router(config-subif)#commit
Router(config-subif)#exit
Router(config)#interface Bundle-Ether2.10 l2transport
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#rewrite ingress tag pop 1 symmetric
Router(config-subif)#commit
Router(config-subif)#root
Router(config)#interface BVI10
Router(config-if)#ipv4 address 10.0.0.10 255.255.255.0
Router(config-if)exit
```

```
Router(config)#interface BVI10
Router(config-if)#ipv4 address 10.0.0.10 255.255.255.0
Router(config-if)#exit
Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg1
Router(config-l2vpn-bg)#bridge-domain bd-10
Router(config-l2vpn-bg-bd)#interface Bundle-Ether1.10
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)#interface Bundle-Ether2.10
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)#routed interface BVI10
Router(config-l2vpn-bg-bd-bvi)#commit
```

Configure EVPN based single-active multi-homing on PE Routers.

PE1 Configuration:

```
Router#configure
Router(config)#interface Bundle-Ether1
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Bundle-Ether2
Router(config-if)#no shutdown
Router(config)#exit
Router(config)#interface Bundle-Ether1.10 l2transport
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#rewrite ingress tag pop 1 symmetric
Router(config-subif)#exit
Router(config)#evpn
Router(config-evpn)#evi 100
Router(config-evpn-instance)#advertise-mac
Router(config-evpn-instance-mac)#exit
Router(config-evpn-instance)#exit
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment
Router(config-evpn-ac-es)#identifier type 0 36.37.00.00.00.00.00.11.00
Router(config-evpn-ac-es)#load-balancing-mode single-active
Router(config-evpn-ac-es)#root
Router(config)#l2vpn
Router(config-l2vpn)#bridge group 100
Router(config-l2vpn-bg)#bridge-domain 100
Router(config-l2vpn-bg-bd)#interface Bundle-Ether2.10
Router(config-l2vpn-bg-bd)#evi 100
Router(config-l2vpn-bg-bd-evi)#commit
```

PE2 Configuration:

```
Router#configure
Router(config)#interface Bundle-Ether1
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Bundle-Ether2
Router(config-if)#no shutdown
Router(config)#exit
Router(config)#interface Bundle-Ether2.10 l2transport
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#rewrite ingress tag pop 1 symmetric
Router(config-subif)#exit
Router(config)#evpn
Router(config-evpn)#evi 100
Router(config-evpn-instance)#advertise-mac
Router(config-evpn-instance-mac)#exit
```

```
Router(config-evpn-instance)#exit
Router(config-evpn)#interface Bundle-Ether2
Router(config-evpn-ac)#ethernet-segment 0 36.37.00.00.00.00.00.11.00
Router(config-evpn-ac-es)#load-balancing-mode single-active
Router(config-evpn-ac-es)#root
Router(config)#l2vpn
Router(config-l2vpn)#bridge group 100
Router(config-l2vpn-bg)#bridge-domain 100
Router(config-l2vpn-bg-bd)#interface Bundle-Ether2.10
Router(config-l2vpn-bg-bd)#evi 100
Router(config-l2vpn-bg-bd-evi)#exit
```

Running Configuration

This section shows the single-active running configuration.

```
/* CE1 Configuration */
interface Bundle-Ether1
!
interface Bundle-Ether1.10 l2transport
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
!
interface Bundle-Ether2
!
interface Bundle-Ether2.10 l2transport
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
!
interface Loopback0
 ipv4 address 200.0.0.7 255.255.255.255
!
interface MgmtEth0/RSP0/CPU0/0
 ipv4 address dhcp
!
interface BVI10
 description "Host-1 IP"
 ipv4 address 10.0.0.10 255.255.255.0
!
interface HundredGigE0/0/0/0
 bundle id 1 mode active
!
interface HundredGigE0/0/0/1
 description "Link to Leaf-2"
 bundle id 2 mode active
!
l2vpn
 bridge group bg1
  bridge-domain bd-10
   interface Bundle-Ether1.10
   !
   interface Bundle-Ether2.10
   !
   routed interface BVI10
   !
  !


/* PE1 Configuration */
evpn
 evi 100
  advertise-mac
  !
```

```
 !
 interface Bundle-Ether1
  ethernet-segment
  identifier type 0 36.37.00.00.00.00.00.11.00
   load-balancing-mode single-active
  !
 !
!
l2vpn
 bridge group 100
  bridge-domain 100
   interface Bundle-Ether1.10
   !
   evi 100
   !
  !
 !
!
commit
root
exit


/* PE2 Configuration */
evpn
 evi 100
  advertise-mac
  !
 !
 interface Bundle-Ether2
  ethernet-segment
   identifier type 0 36.37.00.00.00.00.00.11.00
   load-balancing-mode single-active
  !
 !
!
l2vpn
 bridge group 100
  bridge-domain 100
   interface Bundle-Ether2.10
   !
   evi 100
   !
  !
 !
!
```

Verification

The following output shows that the EVPN single-active mode is enabled:

```
Router#show evpn ethernet-segment detail

Legend:
  B   - No Forwarders EVPN-enabled,
  C   - Backbone Source MAC missing (PBB-EVPN),
  RT  - ES-Import Route Target missing,
  E   - ESI missing,
  H   - Interface handle missing,
  I   - Name (Interface or Virtual Access) missing,
  M   - Interface in Down state,
  O   - BGP End of Download missing,
  P   - Interface already Access Protected,
```

```
  Pf  - Interface forced single-homed,
  R   - BGP RID not received,
  S   - Interface in redundancy standby state,
  X   - ESI-extracted MAC Conflict
  SHG - No local split-horizon-group label allocated

Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0036.3700.0000.0000.1100 BE1                               10.1.1.1
                                                           10.2.2.2
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether1
     Interface MAC  : 0008.3302.3208
     IfHandle       : 0x02000160
     State          : Up
     Redundancy     : Not Defined
  ESI type          : 0
     Value          : 36.3700.0000.0000.1100
  ES Import RT      : 3637.0000.0000 (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH, Single-active
     Configured     : Single-active (AApS)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       :
     Mobility-Flush : Count 0, Skip 0, Last n/a
  Peering Details   : 2 Nexthops
     10.1.1.1 [MOD:P:00]
     10.2.2.2 [MOD:P:00]
  Service Carving Results:
     Forwarders     : 1
     Elected        : 1
     Not Elected    : 0
  EVPN-VPWS Service Carving Results:
     Primary        : 0
     Backup         : 0
     Non-DF         : 0
  MAC Flushing mode : STP-TCN
  Peering timer     : 3 sec [not running]
  Recovery timer    : 30 sec [not running]
  Carving timer     : 0 sec [not running]
  Local SHG label   : 24007
  Remote SHG labels : 1
              24007 : nexthop 10.2.2.2
  Access signal mode: Bundle OOS (Default)
```

The following output shows that Bundle-Ether1 is up:

```
Router:PE1#show bundle bundle-ether 1

Bundle-Ether1
  Status:                                  Up
  Local links <active/standby/configured>:   1 / 0 / 1
  Local bandwidth <effective/available>:     100000000 (100000000) kbps
  MAC address (source):                      0008.3532.0137 (Chassis pool)
  Inter-chassis link:                        No
  Minimum active links / bandwidth:          1 / 1 kbps
  Maximum active links:                      64
  Wait while timer:                          2000 ms
  Load balancing:
    Link order signaling:                    Not configured
```

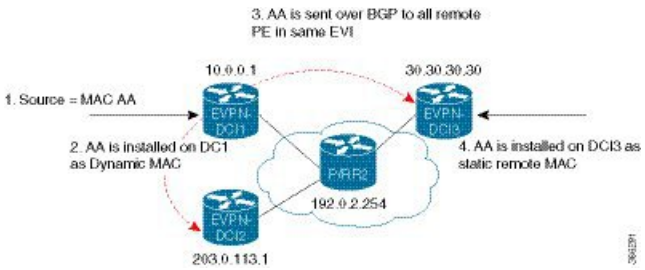
```
  Hash type:                                Default
  Locality threshold:                       None
LACP:                                       Operational
  Flap suppression timer:                   Off
  Cisco extensions:                         Disabled
  Non-revertive:                            Disabled
mLACP:                                      Not configured
IPv4 BFD:                                   Not configured
IPv6 BFD:                                   Not configured

Port                  Device           State        Port ID         B/W, kbps
--------------------  ---------------  -----------  --------------  ----------
Hu0/0/0/0             Local            Active       0x8000, 0x0001   100000000
    Link is Active
```

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.

Note A device can contain up to 128K MAC address entries. A bridge domain on a device can contain up to 64K MAC address entries.

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:

Note From Release 7.4.1 Control word is enabled by default. If the **control-word-disable** command is not configured, ensure to configure it under EVPN or EVI configuration mode before an upgrade to avoid inconsistent behaviour with routers running before Release 7.4.2.

If you want to enable **control-word** command for EVPN Bridging feature, then you must configure it only when both the endpoints run Release 7.4.1 or later.

If you want to disable control word command, use **control-word-disable** before Release 7.8.1, it needed a router to reload to take effect.

Note The router does not support flow-aware transport (FAT) pseudowire.

```
/* 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-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/1
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 l2vpn evpn
```

Supported Modes for EVPN Software MAC Learning

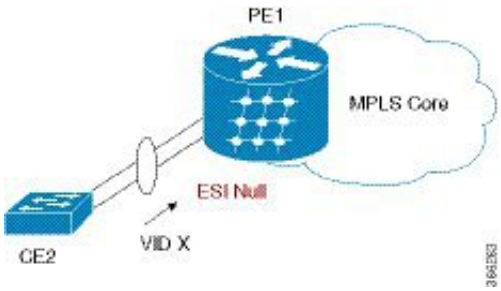
The following are the modes in which EVPN Software MAC Learning is supported:

- 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# configure
RP/0/RSP0/CPU0:router#(config)# router bgp 200
RP/0/RSP0/CPU0:router#(config-bgp)# bgp router-id 09.165.200.227
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-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

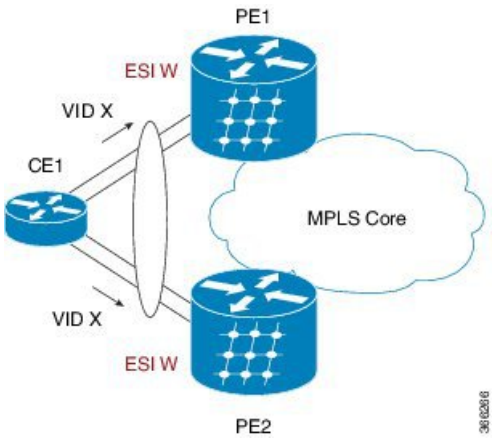
Verify EVPN in single home devices.

```
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
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)# 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 01.11.00.00.00.00.00.00.01


/* 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-nbr)# remote-as 200
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# description MPLS-FACING-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


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

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

Note Configure the same mlacp system priority <id> for both the dual homed PE routers to enable all-active load balancing.

Running Configuration

```
l2vpn
bridge group EVPN_ALL_ACTIVE
 bridge-domain EVPN_2001
 interface Bundle-Ether1
 !
 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 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/CPU0:router# show evpn ethernet-segment interface Bundle-Ether 1 carvin$

Ethernet Segment Id        Interface  Nexthops
-------- ----------        --------   --------
0100.211b.fce5.df00.0b00   BE1       10.10.10.10
 209.165.201.1
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
 Remote SHG labels : 1
  38216 : nexthop 209.165.201.1
```

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
 evi: 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
...
Neighbor        Spk   AS     MsgRcvd  MsgSent  TblVer    InQ  OutQ  Up/Down  St/PfxRcd
209.165.200.225  0    200     216739  229871   200781341  0    0     3d00h   348032
209.165.201.30   0    200     6462962 4208831  200781341  10   0     2d22h   35750
```

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

```
RP/0/RSP0/CPU0:router# show l2vpn mac mac all location 0/6/CPU0

 Topo ID     Producer     Next Hop(s)      Mac Address       IP Address
 --------    --------     -----------      --------------    ----------
     1112    0/6/CPU0     Te0/6/0/1.36001  00a3.0001.0001
```

Verify the MAC updates to the L2FIB table in a route switch processor (RSP).

```
RP/0/RSP0/CPU0:router# show l2vpn mac mac all location 0/6/CPU0

 Topo ID     Producer     Next Hop(s)      Mac Address       IP Address
 --------    --------     -----------      --------------    ----------
     1112    0/6/CPU0     Te0/6/0/1.36001  00a3.0001.0001
```

Verify the summary information for the MAC address.

```
RP/0/RP0/CPU0:router# show l2vpn forwarding bridge-domain EVPN_ALL_ACTIVE:EVPN_2001
mac-address location 0/6/CPU0


    Mac Address    Type    Learned from/Filtered on    LC learned Resync Age/Last Change
Mapped to
    -------------- ------- -------------------------- ---------- ----------------------
--------------
    00a3.0001.0001 dynamic Te0/6/0/1.36001            N/A        01 Sep 10:09:17
N/A
    0010.0400.0003 dynamic Te0/0/0/10/0.1             N/A        Remotely Aged
N/A
    2000.3000.4000 static  Te0/0/0/10/0.2             N/A        N/A
N/A
```

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

```
RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 mac
VPN-ID     Encap      MAC address    IP address     Nexthop
  Label
---------- ---------- -------------- ------------- ---------------------------------------
 --------
  2001               00a9.2002.0001 ::            10.10.10.10
  34226    <-- Remote MAC
  2001               00a9.2002.0001 ::            209.165.201.30
  34202

  2001               00a3.0001.0001 20.1.5.55     TenGigE0/6/0/1.36001
  34203    <-- Local MAC


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

EVI     MAC address      IP address  Nexthop      Label
----    --------------   ----------  -------      -----
2001    00a9.2002.0001   ::          10.10.10.10  34226

2001    00a9.2002.0001   ::          209.165.201.30  34202

 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           0 i <------ locally learnt MAC
*>i[2][0][48][00a9.2002.00be][0]/104
                         10.10.10.10 100    0 i <----- remotely learnt MAC
* i 209.165.201.30 100 0 i
```

EVPN Out of Service

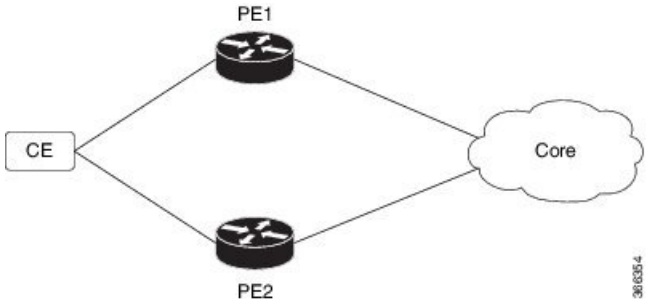
The EVPN Out of Service feature enables you to control the state of bundle interfaces that are part of an Ethernet segment that have Link Aggregation Control protocol (LACP) configured. This feature enables you to put a node out of service (OOS) without having to manually shutdown all the bundles on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

Note 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.

```
/* 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

```
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    : 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    : 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                 : FALSE
      startup-cost-in timer   : Not configured
```

```
/* Verify the startup-cost-in timer 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    : 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 node cost-out            : TRUE
        startup-cost-in timer : 6000
```

CFM Support for EVPN

Ethernet Connectivity Fault Management (CFM) is a service-level OAM protocol that provides tools for monitoring and troubleshooting end-to-end Ethernet services per VLAN. This includes proactive connectivity monitoring, fault verification, and fault isolation. CFM can be deployed in an EVPN network. You can monitor the connections between the nodes using CFM in an EVPN network.

Restrictions

CFM for EVPN is supported with the following restrictions:

- Starting with Cisco IOS XR 7.4.1 release, CFM over EVPN services are not supported in N540-24Q8L2DD-SYS router.
- In an active-active multi-homing scenario, when monitoring the connectivity between a multi-homed CE device and the PE devices to which it is connected, CFM can only be used across each individual link between a CE and a PE. Attempts to use CFM on the bundle between CE and PE devices cause sequence number errors and statistical inaccuracies.
- There is a possibility of artefacts in loopback and linktrace results. Either a loopback or linktrace may report multiple results for the same instance, or consecutive instances of a loopback and linktrace between the same two endpoints may produce different results.

For more information about Ethernet Connectivity Fault Management (CFM), refer to the *Configuring Ethernet OAM* chapter in the *Interface and Hardware Component Configuration Guide for Cisco NCS540 Series Routers*.

Control Word Support for ELAN

Table 3: Feature History Table

Feature Name	Release Information	Feature Description

Control-word support for EVPN Bridge-Mode (E-LAN)	Release 7.4.1	Control word is now supported and enabled by default in ELAN mode. If the **control-word-disable** command is not configured, ensure to configure it under EVPN or EVI configuration mode before an upgrade to avoid inconsistent behaviour with routers before this release.
		`Router# configure`
		`Router(config)# evpn`
		`Router(config-evpn)# evi 1`
		`Router(config-evpn-instance)# control-word-disable`
		`// Apply to interop with older releases EVPN ELAN`
		If you want to enable **control-word** command for EVPN Bridging feature, then you must configure it only when both the endpoints run Release 7.4.1 or later.

Note Control word is enabled by default in ELAN mode as well. If the **control-word-disable** command is not configured, ensure to configure it under EVPN or EVI configuration mode before an upgrade to avoid inconsistent behaviour with routers before Release 7.4.1.

If you want to enable **control-word** command for EVPN Bridging feature, then you must configure it only when both the endpoints run Release 7.4.1 or later.

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 (FXC) 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 540 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 540 Series Routers*.

- EVPN Integrated Routing and Bridging (IRB)

  For more information, see *Configure EVPN IRB* chapter in *L2VPN and Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*.

- Native EVPN

  For more information see, *EVPN Features* chapter in *L2VPN and Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*.

All these services are supported only on all-active multihoming scenario.

Configure EVPN 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.

```
/* Configure attachment circuits */
Router# 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-l2vpn)# 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-l2vpn)# 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-l2vpn)# 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-l2vpn)# 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-l2vpn)# 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-l2vpn)# 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-l2vpn)# 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

/* Configure VLAN Aware FXC Service */
Router(config)# l2vpn
Router(config-l2vpn)# flexible-xconnect-service vlan-aware evi 24001
Router(config-l2vpn-fxs-va)# interface Bundle-Ether22001.12
Router(config-l2vpn-fxs-va)# interface Bundle-Ether22001.13
Router(config-l2vpn-fxs-va)# interface Bundle-Ether22001.14
Router(config-l2vpn-fxs-va)# commit

/* Configure Local Switching - Local switching is supported only on VLAN-aware FXC */
PE1
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# flexible-xconnect-service vlan-aware evi 31400
Router(config-l2vpn-fxs-va)# interface Bundle-Ether22001.1400
Router(config-l2vpn-fxs-va)# interface Bundle-Ether23001.1400
Router(config-l2vpn-fxs-va)# commit
Router(config-l2vpn-fxs)# exit
PE2
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# flexible-xconnect-service vlan-aware evi 31401
Router(config-l2vpn-fxs-va)# interface Bundle-Ether22001.1401
Router(config-l2vpn-fxs-va)# interface Bundle-Ether23001.1401
Router(config-l2vpn-fxs-va)# commit
Router(config-l2vpn-fxs)# exit

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

Router# configure
Router(config)# interface Bundle-Ether22001.11 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 11
Router(config-l2vpn-subif)# rewrite ingress tag pop 2 symmetric
Router(config-l2vpn-subif)# commit
Router(config-l2vpn-subif)# exit

Router# configure
Router(config)# interface Bundle-Ether22001.21 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q  21
Router(config-l2vpn-subif)# rewrite ingress tag pop 2 symmetric
Router(config-l2vpn-subif)# commit
Router(config-l2vpn-subif)# exit

Router# configure
Route(config)# l2vpn
Router(config-l2vpn)# xconnect group xg22001
Router(config-l2vpn-xc)# p2p evpn-vpws-mclag-22001
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether22001.11
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 22101 target 220101 source 220301
Router(config-l2vpn-xc-p2p-pw)# commit
Router(config-l2vpn-xc-p2p-pw)# exit

Router # configure
Router (config)# l2vpn
Router (config-l2vpn)# bridge group native_evpn1
Router (config-l2vpn-bg)#  bridge-domain bd21
Router (config-l2vpn-bg-bd)# interface Bundle-Ether22001.21
Router (config-l2vpn-bg-bd-ac)# routed interface BVI21
```

```
Router (config-l2vpn-bg-bd-bvi)# evi 22021
Router (config-l2vpn-bg-bd-bvi)# commit
Router (config-l2vpn-bg-bd-bvi)# exit

/* Configure Native EVPN */
Router # configure
Router (config)# 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)# exit
Router (config-evpn)# 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)# 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)# exit
Router (config-evpn-evi)# advertise-mac
Router (config-evpn-evi)# 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
encapsulation 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.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
  !
   evi 22101
    bgp
      route-target import 64:22101
      route-target export 64:22101
    !
  evi 22021
   bgp
     route-target import 64:22021
     route-target export 64:22021
   !
   advertise-mac
 !
 evi 22022
  bgp
   route-target import 64:22022
   route-target export 64:22022
  !
   advertise-mac
 !
```

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, (PP) = 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, (PP) = 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
              AC:   BE22001.14    UP
-------------------------------------------------------------------------------------------


Router# show l2vpn xconnect group xg22001 xc-name evpn-vpws-mclag-22001
Fri Sep 1 17:28:58.259 UTC
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
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 Single-Flow-Active Multihoming Load-Balancing Mode

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Single-Flow Active (for Access Rings) - VPNv4 Hosts	Release 7.4.1	This feature extends the current implementation of EVPN Single-Flow-Active Multihoming Load-Balancing Mode, on Cisco routers with VPNv4 routes.

Feature Name	Release Information	Feature Description
EVPN Single-Flow-Active Multihoming Load-Balancing Mode	Release 7.3.1	This feature introduces EVPN Single-Flow-Active multihoming mode to connect PE devices in an access network that run Layer 2 access gateway protocols. In this mode, only the PE that first advertises the host MAC address in a VLAN forwards the traffic in a specific flow. When the primary link fails, the traffic quickly switches to the standby PE that learns the MAC address from the originated path, thereby providing fast convergence. A keyword, **single-flow-active** is added to the **load-balancing-mode** command.

In a ring topology, only one of the PEs, which is the active PE, sends and receives the traffic to prevent a traffic loop. When the link to the active PE fails, the traffic switches over to the standby PE. Traffic switchover takes a while because the standby PE has to learn the MAC addresses of the connected hosts. There's a traffic loss until the traffic switch over happens.

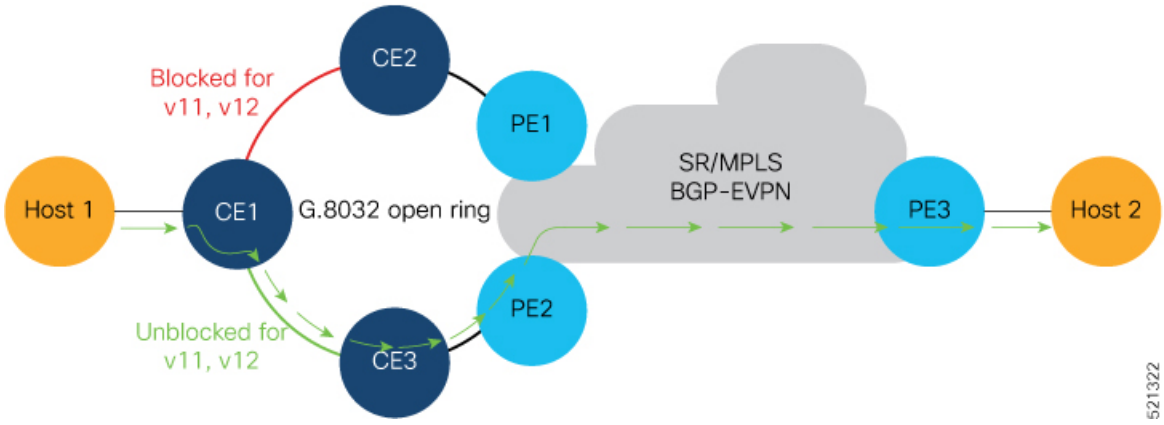
The EVPN Single-Flow-Active multihoming mode connects PE devices in an access network, and in the event of active link failure the switchover happens immediately and reduces the traffic loss.

Both active and standby PEs learn the MAC addresses of the connected host. The PE that learns the MAC address of the host directly is called the Primary (active) PE. The primary PE advertises the learnt MAC addresses to the peer PE, which is referred as standby PE. As the standby PE learns the MAC address of the host through the active PE, this learnt path is referred to as the reoriginated path.

When the primary link fails, the convergence happens fast and the traffic is sent through the standby PE (reoriginated path).

Let us understand how EVPN single flow-active mode helps in fast convergence:

- In this topology, the access network devices are connected through a ring topology. The access network uses Layer-2 gateway protocols such as G.8032, MPLS-TP, STP,REP-AG or MSTP-AG to prevent traffic loop due to continuous flooding.The access protocols are not supported on Cisco NC57 line cards, but only xconnect is supported in access network.



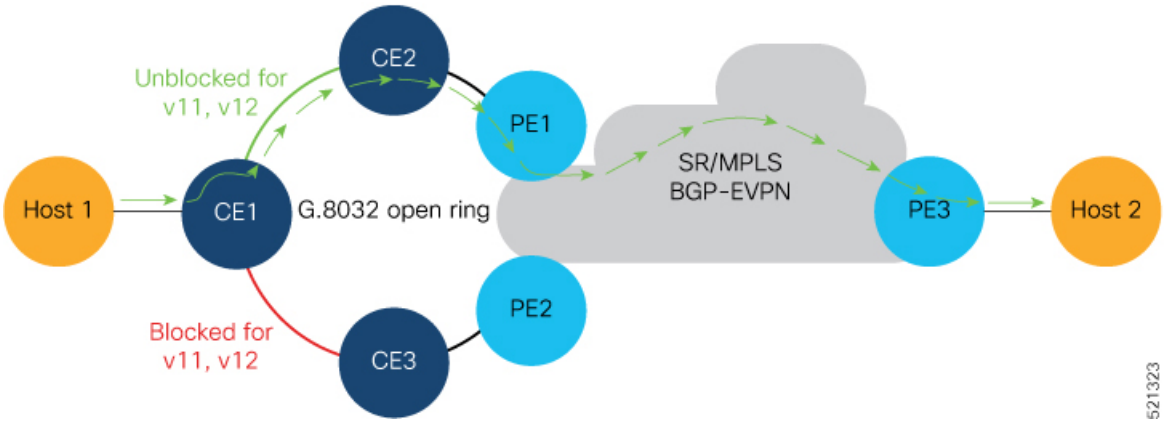
- Host 1 is connected to CE1.
- CE1 is connected to both PE1 and PE2, thus is multihomed.
- PE1 and PE2 are Multihoming devices.
- Both PE1 and PE2 is configured with the same non-zero Ethernet Segment ID (ESI) number `0 36.37.00.00.00.00.00.11.00` for the bundle interface to enable multihoming of the host (CE1).
- PE1 and PE2 belongs to te same VLAN and hence configured with the same EVPN instance (EVI) 100.

Traffic Flow

- Consider a traffic flow from Host 1 to Host 2. The traffic is sent from Host 1 to CE1.
- In this ring topology, the link between CE1 to CE2 is in the blocked state; the link between CE1 to CE3 is in the forwarding state. Hence, CE1 sends the traffic to PE2 through CE3.
- PE2 first learns the MAC address of Host1 through CE1. PE2 advertises the learnt MAC address to the peering PE1.
- As PE2 has learnt the MAC address directly from Host 1, and acts as an active PE.
- The PE which originates the MAC route due to access learning sets the default BGP local preference attribute value to 100.
- PE1 learns the MAC address from PE2 and acts as a stand-by PE. As PE1 gets the reoriginated MAC route from PE2, PE1 sets the BGP local preference attribute value to 80.
- The PE that has the higher local preference always sends and receives the traffic. Thus PE1 sends the traffic to PE3. PE3 sends the traffic to Host 2.

Failure Scenario

When the link between CE1 and CE3 is down or when the link between CE3 and PE2 is down, traffic is sent through PE1.



- When the link fails, the link CE1-CE2 changes to the forwarding state.
- PE1 learns the MAC address of Host 1 directly and advertises the learnt MAC address to PE2.
- PE1 sends the traffic to Host 2 through the remote PE3 with a BGP local preference value of 100.
- PE3 sends and receives the traffic from PE1 until the access link between CE1 and CE2 changes to the blocked state.

Restrictions

Single-Flow Active is not supported for EVPN VPWS.

Configuration Example

- Configure both PE1 and PE2 with the same EVI of 100.
- Configure both PE1 and PE2 with the same ESI 0 36.37.00.00.00.00.00.11.01.

Perform these tasks on both PE1and PE2.

```
/* Configure advertisement of MAC routes */
Router# configure
Router(config)# evpn
Router(config-evpn)# evi 100
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance-mac)# root

/* Configure single-flow-active load-balancing mode */
Router(config)# evpn
Router(config-evpn)# interface bundle-ether 1
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 36.37.00.00.00.00.00.11.01
Router(config-evpn-ac-es)# load-balancing-mode single-flow-active
Router(config-evpn-ac-es)# convergence
Router(config-evpn-ac-es-conv)# mac-mobility
Router(config-evpn-ac-es-conv)# exit

Router(config-evpn-ac-es)# root

/* Configure bridge domain and associating the evi to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group 100
```

```
Router(config-l2vpn-bg)# bridge-domain 100
Router(config-l2vpn-bg-bd)# interface Bundle-Ether1.2
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)# evi 100
Router(config-l2vpn-bg-bd-evi)# root
Router(config)# interface Bundle-Ether1.2 l2transport
Router(config-l2vpn-subif)#encapsulation dot1q 2
Router(config-l2vpn-subif)#commit
```

Running Configuration

```
evpn
 evi 100
  advertise-mac
  !
 !
 interface Bundle-Ether1
  ethernet-segment
   identifier type 0 36.37.00.00.00.00.00.11.01
   load-balancing-mode single-flow-active
  convergence
      mac-mobility
   !
  !
 !
!
l2vpn
  bridge group 100
   bridge-domain 100
    interface Bundle-Ether1
   !
   evi 100
   !
  !
 !
 interface Bundle-Ether1.2 l2transport
  encapsulation dot1q 2
!
!
```

Verification

Verify the Ethernet Segment Status:

- Verify that the Ethernet Segment Id is the same as that you have configured: In this example, you notice that the ESI on PE1 is 0 36.37.00.00.00.00.00.11.01.
- Verify that the Single-flow-active mode is enabled in the Topology section.

```
Router#show  evpn  ethernet-segment  interface  be 1 detail
Legend:
B    - No Forwarders EVPN-enabled,
C    - MAC missing (Backbone S-MAC PBB-EVPN / Grouping ES-MAC vES),
RT   - ES-Import Route Target missing,
E    - ESI missing,
H    - Interface handle missing,
I    - Name (Interface or Virtual Access) missing,
M    - Interface in Down state,
O    - BGP End of Download missing,
P    - Interface already Access Protected,
Pf   - Interface forced single-homed,
```

```
R   - BGP RID not received,
S   - Interface in redundancy standby state,
X   - ESI-extracted MAC Conflict
SHG - No local split-horizon-group label allocated
Hp  - Interface blocked on peering complete during HA event
Rc  - Recovery timer running during peering sequence

Ethernet Segment Id            Interface                           Nexthops
0 36.37.00.00.00.00.00.11.01   BE1                                 172.16.0.4
                                                                   172.16.0.5
ES to BGP Gates   : Ready
ES to L2FIB Gates : P
Main port         :
Interface name    : Bundle-Ether1
Interface MAC     : b0a6.51e5.00dd
IfHandle          : 0x2000802c
State             : Up
Redundancy        : Not Defined
ESI type          : 0
Value             : 07.0807.0807.0807.0800
ES Import RT      : 0708.0708.0708 (from ESI)
Source MAC        : 0000.0000.0000 (N/A)
Topology          :
Operational     : MH, Single-flow-active
Configured      : Single-flow-active
Service Carving   : Auto-selection
Multicast         : Disabled
Convergence       : MAC-Mobility
Mobility-Flush    : Debounce 1 sec, Count 0, Skip 0
                  : Last n/a
Peering Details   : 2 Nexthops
172.16.0.4 [MOD:P:00:T]
172.16.0.5 [MOD:P:00:T]
Service Carving Synchronization:
Mode              : NONE
Peer Updates      :
172.16.0.4 [SCT: N/A]
172.16.0.5 [SCT: N/A]
Service Carving Results:
Forwarders    : 1
Elected         : 0
Not Elected   : 0
EVPN-VPWS Service Carving Results:
Primary           : 0
Backup            : 0
Non-DF            : 0
MAC Flushing mode: STP-TCN
Peering timer     : 3 sec [not running]
Recovery timer    : 30 sec [not running]
Carving timer     : 0 sec [not running]
HRW Reset timer   : 5 sec [not running]
Local SHG label   : 24007
Remote SHG labels: 1
24010             : nexthop 172.16.0.5
Access signal mode: Bundle OOS (Default)
```

```
Router#show l2vpn protection main-interface
Main Interface ID              # of subIntf Protected  Protect Type
Bundle-Ether1                  2            Yes        ERP

Instance : 1
State    : FORWARDING

Sub-Intf # : 2
```

```
Flush    # : 6
```

Associated Commands

- **load-balancing-mode**
- **show evpn ethernet-segment**

EVPN Convergence Using NTP Synchronization

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
EVPN Convergence Using NTP Synchronization	Release 7.3.1	This feature leverages the NTP clock synchronization mechanism to handle the transfer of DF role from one edge device to another. In this mechanism, the newly added or recovered PE advertises the Service Carving Timestamp along with the current time to peering PEs. This improves convergence by reducing the time for DF election from three seconds to a few tens of milliseconds. The **show evpn ethernet-segment** command is modified to display the Service-Carving wall clock Timestamp (SCT).

In Ethernet VPN, depending on the load-balancing mode, the Designated Forwarder (DF) is responsible for forwarding Unicast, Broadcast, Unknown Unicast, and Multicast (BUM) traffic to a multihomed Customer Edge (CE) device on a given VLAN on a particular Ethernet Segment (ES).

The DF is selected from the set of multihomed edge devices attached to a given ES. When a new edge router joins the peering group either through failure recovery or booting up of a new device, the DF election process is triggered.

By default, the process of transferring the DF role from one edge device to another takes 3 seconds. The traffic may be lost during this period.

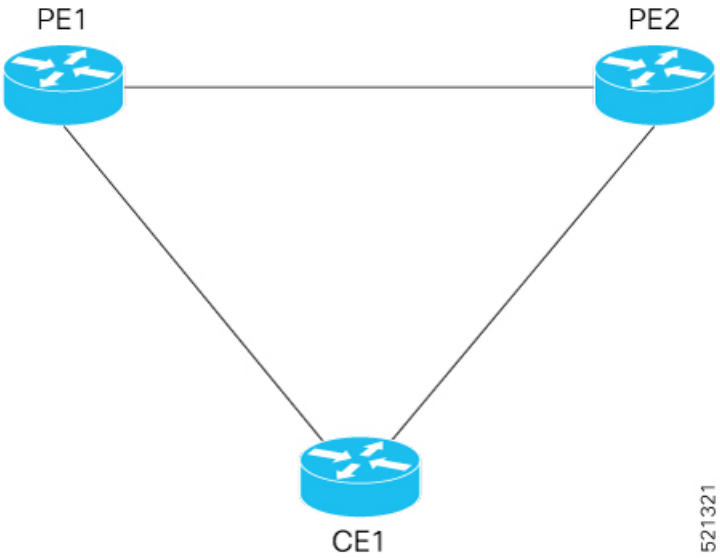
The NTP synchronization mechanism for fast DF election upon recovery leverages the NTP clock synchronization to better align DF events between peering PEs.

If all edge devices attached to a given Ethernet Segment are clock-synchronized with each other using NTP, the default DF election time reduces from 3 seconds to few tens of milliseconds, thereby reducing traffic loss.

Note If the NTP is not synchronized with the NTP server when the EVPN Ethernet Segment interface is coming up, EVPN performs normal DF election.

Let's understand how NTP synchronization works:

Figure 6: EVPN Convergence Using NTP Synchronization



In this topology, CE1 is multihomed to PE1 and PE2.

- PE1 joins the peering group after failure recovery at time (t) = 99 seconds.
- When PE1 joins the peering group, PE1 advertises Route-Type 4 at t = 100 seconds with target Service Carving Time (SCT) value t = 103 seconds to PE2.
- PE2 receives peering Route-Type 4 and learns the DF election time of PE1 to be t =103 seconds.
- If all the peers support NTP, PE2 starts a timer based on the SCT received from PE1 along with a skew value in the Service Carving Time. The skew values are used to eliminate any potential duplicate traffic or loops. Both PE1 and PE2 carves at time t = 103 seconds.

Benefits

- Helps in fast convergence during a primary link recovery
- Supports all the existing load-balancing modes:
  - All-active multihoming
  - Single-active multihoming
  - Port-active multihoming
  - Single-Flow-Active multihoming

Limitations

- All devices attached to a given Ethernet Segment must be configured with NTP. If one of the devices doesn't support NTP clock, the mechanism falls back to default timers.

Verification

Use the **show evpn ethernet-segment** command to view the **Service Carving Time** of the edge device.

For example,

```
Router# show evpn ethernet-segment interface Bundle-Ether200 carving detail

Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0053.5353.5353.5353.5301 BE200                             10.0.0.1
                                                           172.16.0.1
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether200
     Interface MAC  : 2c62.34fd.2485
     IfHandle       : 0x20004334
     State          : Up
     Redundancy     : Not Defined
  ESI type          : 0
     Value          : 53.5353.5353.5353.5301
  ES Import RT      : 8888.8888.8888 (Local)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH, All-active
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       : Reroute
  Peering Details   : 2 Nexthops
     91.0.0.10 [MOD:P:00:T]
     91.0.0.30 [MOD:P:7fff:T]
  Service Carving Synchronization:
     Mode           : NTP_SCT
     Peer Updates   :
               10.0.0.1 [SCT: 2020-10-16 00:28:22:559418]
               10.0.0.3 [SCT: 2020-10-22 17:46:36:587875]
  Service Carving Results:
     Forwarders     : 128
     Elected        : 64

     Not Elected    : 64
```

Associated Commands

- **Show evpn ethernet-segment**

EVPN MPLS Seamless Integration with VPLS

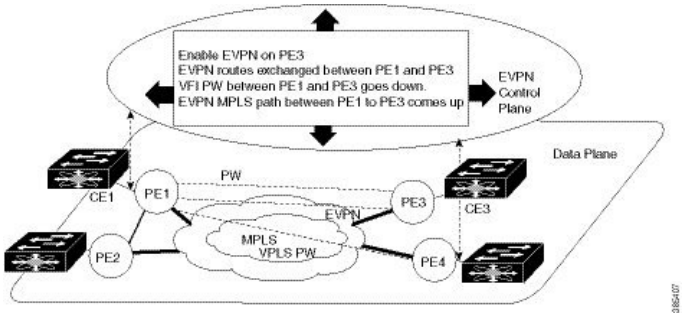
Migrate VPLS Network to EVPN Network through Seamless Integration

In EVPN network, VPN instances are identified by EVPN instance ID (EVI-ID). Similar to other L2VPN technologies, EVPN instances are also associated with route-targets and route-distinguisher. EVPN uses control plane for learning and propagating MAC unlike traditional VPLS, where MAC is learnt in the data plane (learns using "flood and learn technique"). In EVPN, MAC routes are carried by MP-BGP protocol. In

EVPN enabled PEs, PEs import the MAC route along with the label to their respective EVPN forwarding table only if their route targets (RTs) match. An EVPN PE router is capable of performing VPLS and EVPN L2 bridging in the same VPN instance. When both EVPN and BGP-AD PW are configured in a VPN instance, the EVPN PEs advertise the BGP VPLS auto-discovery (AD) route as well as the BGP EVPN Inclusive Multicast route (type-3) for a given VPN Instance. Route type-3 referred to as ingress replication multicast route, is used to send broadcast, unknown unicast, and multicast (BUM) traffic. Other remote PEs import type-3 routes for the same VPN instance only if the sending PE RTs match with their configured RT. Thus, at the end of these route-exchanges, EVPN capable PEs discover all other PEs in the VPN instance and their associated capabilities. The type-3 routes used by PE to send its BUM traffic to other PEs ensure that PEs with the same RTs receive the BUM traffic. EVPN advertises the customer MAC address using type-2 route.

EVPN MPLS Seamless Integration with VPLS allows you to upgrade the VPLS PE routers to EVPN one by one without any network service disruption. Consider the following topology where PE1, PE2, PE3, and PE4 are interconnected in a full-meshed network using VPLS PW.

Figure 7: EVPN MPLS Seamless Integration with VPLS



The EVPN service can be introduced in the network one PE node at a time. The VPLS to EVPN migration starts on PE1 by enabling EVPN in a VPN instance of VPLS service. As soon as EVPN is enabled, PE1 starts advertising EVPN inclusive multicast route to other PE nodes. Since PE1 does not receive any inclusive multicast routes from other PE nodes, VPLS pseudo wires between PE1 and other PE nodes remain active. PE1 keeps forwarding traffic using VPLS pseudo wires. At the same time, PE1 advertises all MAC address learned from CE1 using EVPN route type-2. In the second step, EVPN is enabled in PE3. PE3 starts advertising inclusive multicast route to other PE nodes. Both PE1 and PE3 discover each other through EVPN routes. As a result, PE1 and PE3 shut down the pseudo wires between them. EVPN service replaces VPLS service between PE1 and PE3. At this stage, PE1 keeps running VPLS service with PE2 and PE4. It starts EVPN service with PE3 in the same VPN instance. This is called EVPN seamless integration with VPLS. The VPLS to EVPN migration then continues to remaining PE nodes. In the end, all four PE nodes are enabled with EVPN service. VPLS service is completely replaced with EVPN service in the network. All VPLS pseudo wires are shut down.

Configure EVPN on the Existing VPLS Network

Perform the following tasks to configure EVPN on the existing VPLS network.

- Configure L2VPN EVPN address-family
- Configure EVI and corresponding BGP route-targets under EVPN configuration mode
- Configure EVI under a bridge-domain

See EVI Configuration Under L2VPN Bridge-Domain, on page 45 section for how to migrate various VPLS-based network to EVPN.

Configure L2 EVPN Address-Family

Perform this task to enable EVPN address family under both BGP and participating neighbor.

Configuration Example

```
Router# configure
Router(config)#router bgp 65530
Router(config-bgp)#nsr
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#bgp router-id 200.0.1.1
Router(config-bgp)#address-family l2vpn evpn
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor 200.0.4.1
Router(config-bgp-nbr)#remote-as 65530
Router(config-bgp-nbr)#update-source Loopback0
Router(config-bgp-nbr)#address-family l2vpn evpn
Router(config-bgp-nbr-af)#commit
```

Running Configuration

```
configure
 router bgp 65530
  nsr
  bgp graceful-restart
  bgp router-id 200.0.1.1
  address-family l2vpn evpn
  !
  neighbor 200.0.4.1
   remote-as 65530
   update-source Loopback0
   address-family l2vpn evpn
   !
 !
```

Configure EVI and Corresponding BGP Route Target under EVPN Configuration Mode

Perform this task to configure EVI and define the corresponding BGP route targets. Also, configure advertise-mac, else the MAC routes (type-2) are not advertised.

Configuration Example

```
Router# configure
Router(config)#evpn
Router(config-evpn)#evi 1
Router(config-evpn-evi-bgp)#bgp
Router(config-evpn-evi-bgp)#table-policy spp-basic-6
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
Router(config-evpn-evi)#commit
```

Running Configuration

```
configure
 evpn
  evi
   bgp
    table-policy spp-basic-6
    route-target import 100:6005
    route-target export 100:6005
    !
   advertise-mac
   !
  !
 !
```

Configure EVI under a Bridge Domain

Perform this task to configure EVI under the corresponding L2VPN bridge domain.

Configuration Example

```
Router# configure
Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg1
Router(config-l2vpn-bg)#bridge-domain bd1
Router(config-l2vpn-bg-bd)#interface GigabitEthernet0/0/0/0
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)#evi 1
Router(config-l2vpn-bg-bd-evi)#exit
Router(config-l2vpn-bg-bd)#vfi v1
Router(config-l2vpn-bg-bd-vfi)#neighbor 10.1.1.2 pw-id 1000
Router(config-l2vpn-bg-bd-vfi-pw)#mpls static label local 20001 remote 10001
Router(config-l2vpn-bg-bd-vfi-pw)#commit
```

Running Configuration

```
configure
 l2vpn
  bridge group bg1
   bridge-domain bd1
    interface GigabitEthernet0/0/0/0
    !
    evi 1
    !
   vfi v1
    neighbor 10.1.1.2 pw-id 1000
     mpls static label local 20001 remote 10001
     !
    !
    evi 1
 !
```

EVI Configuration Under L2VPN Bridge-Domain

The following examples show EVI configuration under L2VPN bridge-domain for various VPLS-based networks:

MPLS Static Labels Based VPLS

```
l2vpn
 bridge group bg1
  bridge-domain bd-1-1
   interface GigabitEthernet0/0/0/0
   !
   vfi vfi-1-1
    neighbor 200.0.2.1 pw-id 1200001
     mpls static label local 20001 remote 10001
    !
    neighbor 200.0.3.1 pw-id 1300001
     mpls static label local 30001 remote 10001
    !
    neighbor 200.0.4.1 pw-id 1400001
     mpls static label local 40001 remote 10001
    !
  !
  evi 1
!
```

AutoDiscovery BGP and BGP Signalling Based VPLS

```
l2vpn
bridge group bg1
bridge-domain bd-1-2
   interface GigabitEthernet0/0/0/2
   !
   vfi vfi-1-2
    vpn-id 2
    autodiscovery bgp
     rd 101:2
     route-target 65530:200
     signaling-protocol bgp
      ve-id 11
      ve-range 16
     !
    !
   evi 2
  !
```

Targeted LDP-Based VPLS

```
bridge-domain bd-1-4
   interface GigabitEthernet0/0/0/4
   !
   vfi vfi-1-4
    neighbor 200.0.2.1 pw-id 1200004
    !
    neighbor 200.0.3.1 pw-id 1300004
    !
    neighbor 200.0.4.1 pw-id 1400004
```

```
 !
evi 3
!
```

Verify EVPN Configuration

Use the following commands to verify EVPN configuration and MAC advertisement. Verify EVPN status, AC status, and VFI status.

- show l2vpn bridge-domain
- show evpn summary
- show bgp rt l2vpn evpn
- show evpn evi
- show l2route evpn mac all

Verify the state of the bridge domain, number of ACs, and VFIs.

```
Router#show l2vpn bridge-domain bd-name bd-1-1
Mon Feb 20 21:03:40.244 EST
Legend: pp = Partially Programmed.
Bridge group: bg1, bridge-domain: bd-1-1, id: 0, state: up, ShgId: 0, MSTi: 0
  Aging: 300 s, MAC limit: 4000, Action: none, Notification: syslog
  Filter MAC addresses: 0
  ACs: 1 (1 up), VFIs: 1, PWs: 3 (2 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of EVPNs:
    EVPN, state: up
  List of ACs:
    Gi0/2/0/0.1, state: up, Static MAC addresses: 0, MSTi: 2
  List of Access PWs:
  List of VFIs:
    VFI vfi-1-1 (up)
      Neighbor 200.0.2.1 pw-id 1200001, state: up, Static MAC addresses: 0
      Neighbor 200.0.3.1 pw-id 1300001, state: down, Static MAC addresses: 0
      Neighbor 200.0.4.1 pw-id 1400001, state: up, Static MAC addresses: 0
  List of Access VFIs:
 When PEs are evpn enabled, pseudowires that are associated with that BD will be brought
down. The VPLS BD pseudowires are always up.
```

Verify the number of EVI's configured, local and remote MAC-routes that are advertised.

```
Router#show evpn summary
Mon Feb 20 21:05:16.755 EST
------------------------------
Global Information
------------------------------
Number of EVIs                  : 6
Number of Local EAD Entries     : 0
Number of Remote EAD Entries    : 0
Number of Local MAC Routes      : 4
          MAC                   : 4
          MAC-IPv4              : 0
          MAC-IPv6              : 0
Number of Local ES:Global MAC   : 1
Number of Remote MAC Routes     : 0
          MAC                   : 0
          MAC-IPv4              : 0
```

```
          MAC-IPv6                : 0
Number of Remote SOO MAC Routes   : 0
Number of Local IMCAST Routes     : 4
Number of Remote IMCAST Routes    : 4
Number of Internal Labels         : 0
Number of ES Entries              : 1
Number of Neighbor Entries        : 4
EVPN Router ID                    : 200.0.1.1
BGP ASN                           : 65530
PBB BSA MAC address               : 0026.982b.c1e5
Global peering timer              :      3 seconds
Global recovery timer             :     30 seconds
```

Verify EVPN route-targets.

```
Router#show bgp rt l2vpn evpn
Mon Feb 20 21:06:18.882 EST
EXTCOMM         IMP/EXP
RT:65530:1               1 / 1
RT:65530:2               1 / 1
RT:65530:3               1 / 1
RT:65530:4               1 / 1
Processed 4 entries
```

Verify EVPN MAC routes pertaining to specific VPN instance.

```
Router#show evpn evi vpn-id 1 mac
Mon Feb 20 21:36:23.574 EST

EVI        MAC address    IP address                    Nexthop
Label
---------- -------------- ----------------------------------------
---------------------------------
1      0033.0000.0001      ::                       200.0.1.1                          45106
```

Verify L2 routing.

```
Router#show l2route evpn mac all
Mon Feb 20 21:39:43.953 EST
Topo ID  Mac Address    Prod   Next Hop(s)
-------- -------------- ------ ----------------------------------------
0        0033.0000.0001 L2VPN  200.0.1.1/45106/ME
1        0033.0000.0002 L2VPN  200.0.1.1/45108/ME
2        0033.0000.0003 L2VPN  200.0.1.1/45110/ME
3        0033.0000.0004 L2VPN  200.0.1.1/45112/ME
```

Verifty EVPN route-type 2 routes.

```
Router#show bgp l2vpn evpn route-type 2
Mon Feb 20 21:43:23.616 EST
BGP router identifier 200.0.3.1, local AS number 65530
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 21
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
```

```
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200.0.1.1:1
*>i[2][0][48][0033.0000.0001][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.1.1:2
*>i[2][0][48][0033.0000.0002][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.1.1:3
*>i[2][0][48][0033.0000.0003][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.1.1:4
*>i[2][0][48][0033.0000.0004][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.3.1:1 (default for vrf bd-1-1)
*>i[2][0][48][0033.0000.0001][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.3.1:2 (default for vrf bd-1-2)
*>i[2][0][48][0033.0000.0002][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.3.1:3 (default for vrf bd-1-3)
*>i[2][0][48][0033.0000.0003][0]/104
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.3.1:4 (default for vrf bd-1-4)
*>i[2][0][48][0033.0000.0004][0]/104
                      200.0.1.1                     100      0 i

Processed 8 prefixes, 8 paths
```

Verify inclusive multicast routes and route-type 3 routes.

```
Router#show bgp l2vpn evpn route-type 3
Mon Feb 20 21:43:33.970 EST
BGP router identifier 200.0.3.1, local AS number 65530
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 21
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200.0.1.1:1
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.1.1:2
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.1.1:3
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.1.1:4
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
Route Distinguisher: 200.0.3.1:1 (default for vrf bd-1-1)
```

```
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
*> [3][0][32][200.0.3.1]/80
                      0.0.0.0                                0 i
Route Distinguisher: 200.0.3.1:2 (default for vrf bd-1-2)
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
*> [3][0][32][200.0.3.1]/80
                      0.0.0.0                                0 i
Route Distinguisher: 200.0.3.1:3 (default for vrf bd-1-3)
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
*> [3][0][32][200.0.3.1]/80
                      0.0.0.0                                0 i
Route Distinguisher: 200.0.3.1:4 (default for vrf bd-1-4)
*>i[3][0][32][200.0.1.1]/80
                      200.0.1.1                     100      0 i
*> [3][0][32][200.0.3.1]/80
                      0.0.0.0                                0 i
```

Clear Forwarding Table

To clear an L2VPN forwarding table at a specified location, you can use the **clear l2vpn forwarding table** command. When BVI is present in the bridge domain, you might experience traffic loss during the command execution. Refer the following work-around to resolve such issues.

When you encounter such issues, delete the BVI and roll back the action. As a result, the traffic on the BVI returns to normal state. The following example shows how to delete the BVI and perform roll back action:

```
Router#clear l2vpn forwarding table location 0/0/CPU0
Fri Mar 24 09:34:02.083 UTC
Router(config)#no int BVI100
Router(config)#commit
Router#roll configuration las 1
Wed Dec 16 18:26:52.869 UTC
Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing
```

Note We can also clear the forwarding table by shutting and unshutting the interface.

Hierarchical EVPN Access Pseudowire

Table 6: Feature History Table

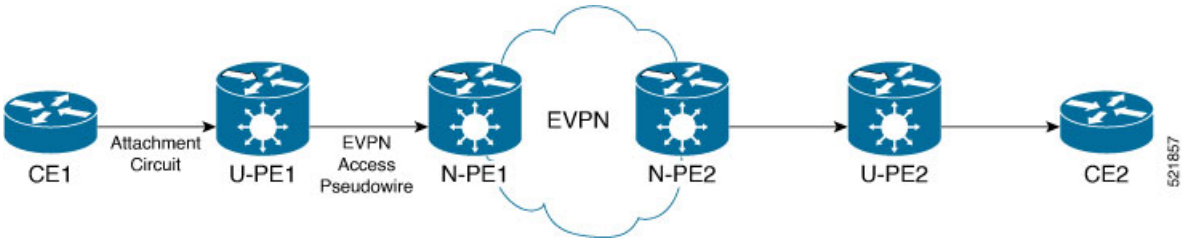
Feature Name	Release Information	Feature Description

Hierarchical EVPN Access Pseudowire	Release 7.6.1	You can configure EVPN VPWS in the access node under the same bridge domain as EVPN in the core to build a PW to the nearest high-end PE that stitches those access circuits using EVPN. This allows the access nodes to leverage the benefits of EVPN. This feature also allows you to reduce the number of pseudowires (PWs) between the network provider edge (N-PE) devices by replacing PE devices with user provider edge (U-PE) and network provider edge (N-PE) devices. This feature prevents signaling overhead and packet replication.

A standard VPN configuration comprises of CE devices and PE devices. With this feature, each PE device is replaced with a user provider edge (U-PE) and network provider edge (N-PE) devices. U-PE devices communicate with the CE devices and N-PE devices on the access side, and N-PE devices communicate with other N-PE devices on the core.

The Hierarchical EVPN Access Pseudowire feature allows you to reduce the number of pseudowires (PWs) between the network provider edge (N-PE) devices. The user provider edge (U-PE) device connects to the N-PE device using EVPN access pseudowire (PW) for each VPN instance. Each CE device is connected to a U-PE device through an attachment circuit.

Hierarchical EVPN Access Pseudowire Topology



In this topology, a user provider edge (U-PE1) device is connected to the CE1 through an attachment circuit. The U-PE1 device transports the CE1 traffic over an EVPN access PW to a network provider edge (N-PE1) device. The N-PE1 is connected with other N-PE2 in an EVPN core. On the N-PE1, the access PW coming from the U-PE1 is much like an AC. The U-PE is not part of the core with the other N-PEs. The N-PE forwards traffic from that access PW to the core PWs that are part of the EVPN core.

Configure Hierarchical EVPN Access Pseudowire

Perform the following task to configure Hierarchical EVPN Access Pseudowire feature on U-PEs and N-PEs.

Configuration Example

```
/* Configure U-PE1 */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group XG1
Router(config-l2vpn-xc)# p2p P1
Router(config-l2vpn-xc-p2p)# interface TenGigE0/0/0/31
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 target 33 source 33
Router(config-l2vpn-xc-p2p-pw)# commit

/* Configure N-PE1 */
```

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group evpn
Router(config-l2vpn-bg)# bridge-domain evpn1
Router(config-l2vpn-bg-bd)# neighbor evpn evi 4 target 33
Router(config-l2vpn-bg-bd)# evi 1
Router(config-l2vpn-bg-bd-evi)# commit
```

Running Configuration

This section shows the Hierarchical EVPN Access Pseudowire running configuration.

```
/* U-PE1 Configuration */
l2vpn
 xconnect group XG1
  p2p P1
   interface TenGigE0/0/0/31 l2transport
   neighbor evpn evi 4 target 33 source 33
   !
!
/* N-PE1 Configuration */
l2vpn
bridge group evpn
  bridge-domain evpn1
   neighbor evpn evi 4 target 33
   evi 1
   !
  !
!
!
```

Verification

Verify the EVPN state, and the list of access PWs. The following is the sample output on N-PE1:

```
Router:N-PE1# show l2vpn bridge-domain bd-name evpn1
Wed Jun 16 09:22:30.328 EDT
Legend: pp = Partially Programmed.
Bridge group: evpn, bridge-domain: evpn1, id: 1, state: up, ShgId: 0, MSTi: 0
  Aging: 300 s, MAC limit: 4000, Action: none, Notification: syslog
  Filter MAC addresses: 0
  ACs: 0 (0 up), VFIs: 0, PWs: 1 (1 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of EVPNs:
    EVPN, state: up
  List of ACs:
  List of Access PWs:
      EVPN 4,33,192.168.0.4, state: up, Static MAC addresses: 0
  List of VFIs:
  List of Access VFIs:
```

Network Convergence using Core Isolation Protection

The Network Convergence using Core Isolation Protection feature allows the router to converge fast when remote links and local interfaces fail. This feature reduces the duration of traffic drop by rapidly rerouting traffic to alternate paths. This feature uses Object Tracking (OT) to detect remote link failure and failure of connected interfaces.

Tracking interfaces can only detect failure of connected interfaces and not failure of a remote router interfaces that provides connectivity to the core. Tracking one or more BGP neighbor sessions along with one or more of the neighbor's address-families enables you to detect remote link failure.

Object Tracking

Object tracking (OT) is a mechanism for tracking an object to take any client action on another object as configured by the client. The object on which the client action is performed may not have any relationship to the tracked objects. The client actions are performed based on changes to the properties of the object being tracked.

You can identify each tracked object by a unique name that is specified by the track command in the configuration mode.

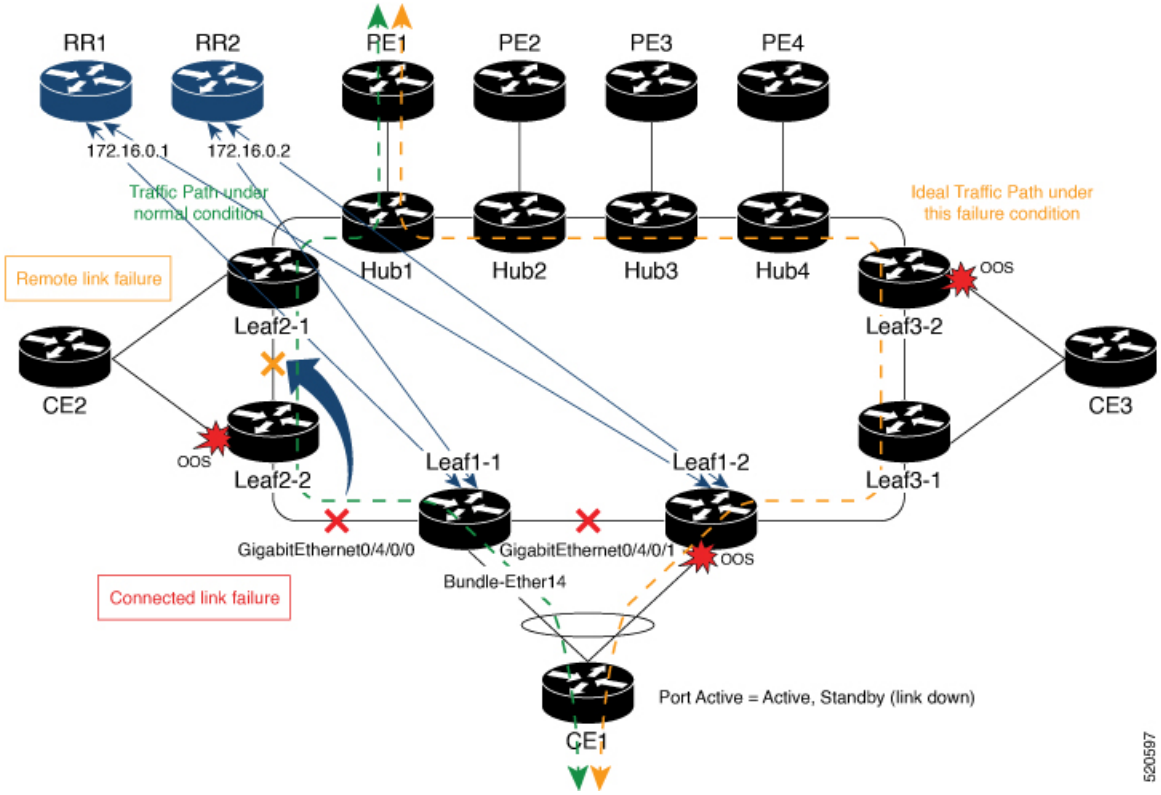
The tracking process receives the notification when the tracked object changes its state. The state of the tracked objects can be up or down.

You can also track multiple objects by a list. You can use a flexible method for combining objects with Boolean logic. This functionality includes:

- Boolean AND function—When a tracked list has been assigned a Boolean AND function, each object defined within a subset must be in an up state, so that the tracked object can also be in the up state.
- Boolean OR function—When the tracked list has been assigned a Boolean OR function, it means that at least one object defined within a subset must also be in an up state, so that the tracked object can also be in the up state.

For more information on OT, see the *Configuring Object Tracking* chapter in the *System Management Configuration Guide for Cisco NCS 540 Series Routers*.

Figure 8: EVPN Convergence Using Core Isolation Protection



Consider a traffic flow from CE1 to PE1. The CE1 can send the traffic either from Leaf1-1 or Leaf1-2. When Leaf1-1 loses the connectivity to both the local links and remote link, BGP sessions to both route reflectors (RRs) are down; the Leaf1-1 brings down the Bundle-Ether14 connected to CE1. The CE1 redirects the traffic from Leaf1-2 to PE1.

You can track the connected interfaces to identify the connected link failures. However, if there is a remote link failure, tracking connected interfaces does not identify the remote link failures. You must track BGP sessions to identify the remote link failure.

Note When you configure the **bgp graceful-restart** command, unconfiguring a neighbor is considered as a non-gr event. This generates a BGP notification to the neighbor before the neighbor is unconfigured.

On the remote router, if the track is configured for this neighbor, the track state is brought down immediately.

However, certain configurations are treated as graceful reset reason and when unconfigured they supress the BGP notification to the neighbor. The route-reflector-client configuration under the neighbor or neighbor address-family is one of the examples.

On the remote router, if the track is configured for this neighbor, the track state is not brought down immediately because a notification is not received.

To overcome this situation, shutdown the neighbor before unconfiguring the neighbor. This generates a BGP notification to the neighbor, and any track configured for the neighbor is brought down immediately.

Configure EVPN Convergence using Core Isolation Protection

A tracked list contains one or more objects. The Boolean expression enables tracking objects using either AND or OR operators. For example, when tracking two interfaces, using the AND operator, up means that *both* interfaces are up, and down means that *either* interface is down.

Note An object must exist before it can be added to a tracked list.

The NOT operator is specified for one or more objects and negates the state of the object.

After configuring the tracked object, you must associate the neighbor or interface whose state must be tracked.

Perform the following tasks to configure EVPN convergence using core isolation protection:

- Configure BGP
- Track the Line Protocol State of an Interface
- Track neighbor adress-family state
- Track objects for both interfaces and neighbors

Configuration Example

In this example, Leaf1-1 brings the down the AC connected to CE1 when:

Both local interfaces GigabitEthernet0/4/0/0 and GigabitEthernet0/4/0/1of Leaf1-1 are down.

OR

Leaf1-1 BGP sessions to both RRs are down.

CE1 re-directs the traffic it was sending to Leaf1-1 to Leaf1-2.

Perform the following tasks on Leaf1-1:

```
/* Configure BGP */
Router# configure
Router(config)# router bgp 100
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 172.16.0.1
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp-nbr-af)# neighbor 172.16.0.1
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp-nbr-af)# commit

/* Track the Line Protocol State of an Interface */
Router# configure
Router(config)# track interface-1
Router(config-track)# type line-protocol state
Router(config-track-line-prot)# interface GigabitEthernet0/4/0/0
Router(config-track-line-prot)#exit
Router(config-track)#exit
Router(config)# track interface-2
Router(config-track)# type line-protocol state
```

```
Router(config-track-line-prot)# interface GigabitEthernet0/4/0/1
Router(config-track-line-prot)#exit
Router(config-track)#exit
Router(config)# track interface-group-1
Router(config-track)# type list boolean or
Router(config-track-list-boolean)# object interface-1
Router(config-track-list-boolean)# object interface-2
Router(config-track-list-boolean)# commit

/* Track neighbor address-family state */
Router# configure
Router(config)# track neighbor-A
Router(config-track)# type bgp neighbor address-family state
Router(config-track-bgp-nbr-af)# address-family l2vpn evpn
Router(config-track-bgp-neighbor)# neighbor 172.16.0.1
Router(config-track-bgp-neighbor)# exit
Router(config-track-bgp-nbr-af)# exit
Router(config-track)# exit
Router(config)# track neighbor-B
Router(config-track)# type bgp neighbor address-family state
Router(config-track-bgp-nbr-af)# address-family l2vpn evpn
Router(config-track-bgp-neighbor)# neighbor 172.16.0.2
Router(config-track-bgp-neighbor)# exit
Router(config-track-bgp-nbr-af)# exit
Router(config-track)# exit
Router(config)# track neighbor-group-1
Router(config-track)# type list boolean or
Router(config-track-list-boolean)# object neighbor-A
Router(config-track-list-boolean)# object neighbor-B
Router(config-track-list-boolean)# commit

/* Track objects for both interfaces and neighbors */
Router# configure
Router(config)# track core-group-1
Router(config-track)# type list boolean and
Router(config-track-list-boolean)# object neighbor-group-1
Router(config-track-list-boolean)# object interface-group-1
Router(config-track-list-boolean)# action
Router(config-track-action)# track-down error-disable interface Bundle-Ether14 auto-recover
Router(config-track-action)# commit
```

Running Configuration

This section shows EVPN convergence using core isolation protection running configuration.

```
router bgp 100
 address-family l2vpn evpn
 !
 neighbor 172.16.0.1
  remote-as 100
  address-family l2vpn evpn
  !
 !
 neighbor 172.16.0.2
  remote-as 100
  address-family l2vpn evpn
  !
 !
!

track interface-1
```

```
 type line-protocol state
  interface GigabitEthernet0/4/0/0
 !
!
track interface-2
 type line-protocol state
  interface GigabitEthernet0/4/0/1
 !
!
track interface-group-1
 type list boolean or
  object interface-1
  object interface-2
 !
!

track neighbor-A
 type bgp neighbor address-family state
  address-family l2vpn evpn
   neighbor 172.16.0.1
  !
 !
!
track neighbor-B
 type bgp neighbor address-family state
  address-family l2vpn evpn
   neighbor 172.16.0.1
  !
 !
!

track neighbor-group-1
 type list boolean or
  object neighbor-A
  object neighbor-B
  !
 !
!
track core-group-1
 type list boolean and
  object neighbor-group-1
  object interface-group-1
 !
 action
  track-down error-disable interface Bundle-Ether14 auto-recover
  !
 !
```

Verification

Verify that you have configured the EVPN convergence using core isolation protection feature successfully.

```
Router# show track
Wed May 27 04:42:11.995 UTC

Track neighbor-A
        BGP Neighbor AF L2VPN EVPN NBR 172.16.0.1 vrf default
        Reachability is UP
                Neighbor Address Reachablity is Up
                BGP Neighbor Address-family state is Up
        4 changes, last change UTC Tue May 26 2020 20:14:33.171

Track neighbor-B
```

```
        BGP Neighbor AF L2VPN EVPN NBR 172.16.0.2 vrf default
        Reachability is UP
                Neighbor Address Reachablity is Up
                BGP Neighbor Address-family state is Up
        4 changes, last change UTC Tue May 26 2020 20:14:27.527

Track core-group-1
        List boolean and is UP
        2 changes, last change 20:14:27 UTC Tue May 26 2020
                object interface-group-1 UP
                object neighbor-group-1 UP

Track interface-1
        Interface GigabitEthernet0/4/0/0 line-protocol
        Line protocol is UP
        2 changes, last change 20:13:32 UTC Tue May 26 2020

Track interface-2
        Interface GigabitEthernet0/4/0/1 line-protocol
        Line protocol is UP
        2 changes, last change 20:13:28 UTC Tue May 26 2020

Track interface-group-1
        List boolean or is UP
        2 changes, last change 20:13:28 UTC Tue May 26 2020
                object interface-2 UP
                object interface-1 UP

Track neighbor-group-1
        List boolean or is UP
        2 changes, last change 20:14:27 UTC Tue May 26 2020
                object neighbor-A UP
                object neighbor-B UP


Router# show track brief
Wed May 27 04:39:19.740 UTC
Track                           Object                                      Parameter
   Value
-------------------------------------------------------------------------------------------------------
neighbor-A                      bgp nbr L2VPN EVPN 172.16.0.1 vrf defau reachability
Up
neighbor-B                      bgp nbr L2VPN EVPN 172.16.0.1 vrf defau reachability
Up
core-group-1                    list                                        boolean and
  Up
interface-1                     interface GigabitEthernet0/4/0/0            line protocol
  Up
interface-2                     interface GigabitEthernet0/4/0/1            line protocol
  Up
interface-group-1               list                                        boolean or
  Up
neighbor-group-1                list                                        boolean or
  Up
-------------------------------------------------------------------------------------------------------

Router# show bgp track
Wed May 27 05:05:51.285 UTC

VRF                   Address-family       Neighbor         Status     Flags

default               L2VPN EVPN           172.16.0.1     UP         0x01
default               L2VPN EVPN           172.16.0.2     UP         0x01
```

```
Processed 2 entries
```

Configurable Recovery Time for EVPN Core Isolation Group

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Configurable Recovery Time for EVPN Core Isolation Group	Release 7.6.1	You can now configure the recovery time for the EVPN core isolation group after the core interfaces recover from a network failure. This functionality is important because post-failure recovery, you can provide sufficient time for the EVPN PE nodes to relearn the MAC addresses and BGP routes received from the remote PEs. There's also time to handle delays in exchanging EVPN routes after recovery. This feature introduces the **core-de-isolation** command under the EVPN Timers configuration mode.

When the core link failure is detected on the PE device, the PE device is isolated from the network and brings down the access interfaces connected to this PE till the core interfaces recover. When the core links recover, the default recovery delay timer begins. The access interfaces become active after the default recover delay timer of 60 seconds expire. The core isolation group recovery delay timer was not user-configurable.

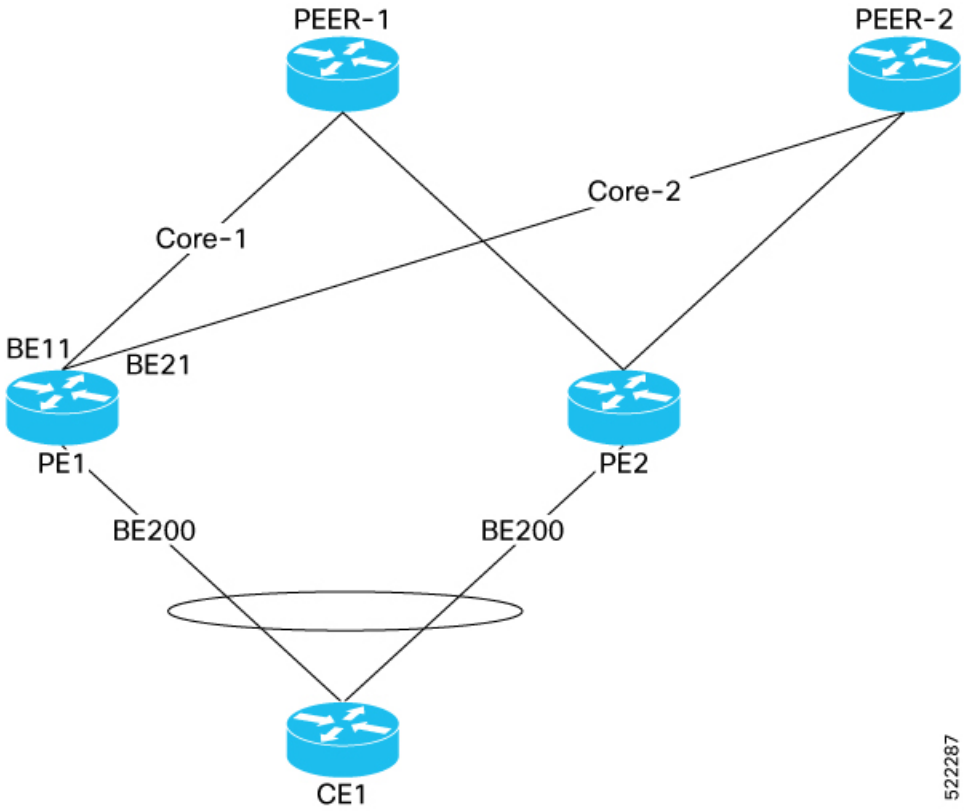
Under scale situations where a network has high MAC addresses, it is observed that the 60 seconds is too short to bring up the access bundle interface as there can be multiple reasons which can delay the exchange of EVPN routes even after the core interfaces have come up.

This feature allows you to configure the core isolation group recovery time to handle delays coming from the core and provides enough time for the EVPN PE nodes to relearn the MAC addresses. You can configure the core isolation group recovery time using the **core-de-isolation** command.

Topology

Consider a topology where CE1 is connected to PE1 and PE2. PE1 and PE2 are running EVPN over the MPLS core network. The core interfaces on PE1 are configured with BE11 and BE22. When the core links of PE1 go down, the EVPN detects the link failure and isolates the PE1 node from the core network, and brings down the access interfaces connected to PE1. This prevents CE1 from sending any traffic to PE1.

When all the core interfaces and BGP sessions come up, PE1 advertises Ethernet A-D Ethernet Segment (ES-EAD) routes again, triggers the service carving, and becomes part of the core network. The access interfaces connected to PE1 from CE1 also come up after the *core-de-isolation* timer value expires.



Configurable Recovery Time for EVPN Core Isolation Group

To enable this feature, configure core interfaces under the EVPN group and associate that group to the Ethernet Segment which is an attachment circuit (AC) attached to the CE.

Perform the following tasks to configure recovery time for EVPN core isolation group:

- Configure EVPN core interfaces on PE1
- Configure *core-de-isolation* timer on PE1
- Configure attachment circuits on CE1

Configuration Example

Configure EVPN core interfaces on PE1.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# group 100
Router(config-evpn-group)# core interface BE11
Router(config-evpn-group)# core interface BE21
Router(config-evpn-group)# commit
```

Configure core-de-isolation timer on PE1.

```
Router# configure
```

```
Router(config)# evpn timers
Router(config-evpn-timers)# core-de-isolation 120
Router(config-evpn-timers)# commit
```

Configure attachment circuits on CE1.

```
/* Configure interface Bundle-Ether200 and associate it to core isolation group 100 */
Router # configure
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether200
Router(config-evpn-ac)# ethernet-segment identifier type 0 11.11.11.11.11.11.11.11.11
Router(config-evpn-ac-es)# bgp route-target 1111.1111.1111
Router(config-evpn-ac-es)# exit
Router(config-evpn-ac)# core-isolation-group 100

/* Configure interface Bundle-Ether201 and associate it to core isolation group 100 */
Router# configure
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether201
Router(config-evpn-ac)# ethernet-segment identifier type 0 11.22.22.22.22.22.22.22.22
Router(config-evpn-ac-es)# bgp route-target 1111.2222.2222
Router(config-evpn-ac-es)# exit
Router(config-evpn-ac)# core-isolation-group 100
```

Running Configuration

This section shows the EVPN core isolation group recovery delay timer running configuration.

```
/* Configure EVPN core interfaces on PE1 */
evpn
 group 100
  core interface Bundle-Ether11
  core interface Bundle-Ether21
!
!
/* Configure core-de-isolation timer on PE1 */
evpn timers
  core-de-isolation 120
!
!
/* Configure attachment circuits on CE1 */
evpn
 interface Bundle-Ether200
  ethernet-segment
   identifier type 0 11.11.11.11.11.11.11.11.11
   bgp route-target 1111.1111.1111
  !
  core-isolation-group 100
!
!
evpn
 interface Bundle-Ether201
  ethernet-segment
   identifier type 0 11.22.22.22.22.22.22.22.22
   bgp route-target 1111.2222.2222
  !
  core-isolation-group 100
!
!
```

Verification

The following output shows that all core interfaces and access interfaces are UP. The *core de-isolation* timer value is configured as 120 seconds, but not running as the core interfaces are UP.

```
Router# show evpn group
EVPN Group: 100

  state: Ready

  Core Interfaces:
     Bundle-Ether11: up
     Bundle-Ether21: up

  Access Interfaces:
     Bundle-Ether200: up
     Bundle-Ether201: up

Router# show evpn summary
-----------------------------
Global Information
-----------------------------
Number of EVIs                    : 141
Number of TEPs                    : 2
Number of Local EAD Entries       : 178
Number of Remote EAD Entries      : 534
Number of Local MAC Routes        : 89
          MAC                     : 89
          MAC-IPv4                : 0
          MAC-IPv6                : 0
Number of Local ES:Global MAC     : 1
Number of Remote MAC Routes       : 0
          MAC                     : 0
          MAC-IPv4                : 0
          MAC-IPv6                : 0
Number of Remote SYNC MAC Routes  : 0
Number of Local IMCAST Routes     : 89
Number of Remote IMCAST Routes    : 178
Number of Internal Labels         : 178
Number of single-home Internal IDs : 0
Number of multi-home Internal IDs  : 0
Number of ES Entries              : 3
Number of Neighbor Entries        : 178
EVPN Router ID                    : 192.168.10.1
BGP ASN                           : 64600
PBB BSA MAC address               : d46a.3599.50d8
Global peering timer              :      3 seconds
Global recovery timer             :     30 seconds
Global carving timer              :      0 seconds
Global MAC postpone timer         :    300 seconds [not running]
Global core de-isolation timer    :    120 seconds [not running]
EVPN services costed out on node  : No
          Startup-cost-in timer   : Not configured
          EVPN manual cost-out    : No
          EVPN Bundle Convergence : No
```

Failure Scenario

The following example shows the failure scenario and how the *core de-isolation* timer works.

Let’s bring down the core interfaces:

```
Router# configure
Router(config)# interface Bundle-Ether11
Router(config-if)# shutdown
Router(config-if)# exit
Router(config)# interface Bundle-Ether21
Router(config-if)# shutdown
Router(config-if)# commit
```

This example shows when the core interfaces are shutdown even the access interfaces are down and the core is isolated.

```
Router# show evpn group

EVPN Group: 100

  state: Isolated

  Core Interfaces:
     Bundle-Ether11: shutdown
     Bundle-Ether21: shutdown

  Access Interfaces:
     Bundle-Ether200: down
     Bundle-Ether201: down
```

This example shows that the *core de-isolation timer* is not yet running because the core interfaces are still down.

```
Router# show evpn summary
------------------------------
Global Information
------------------------------
Number of EVIs                   : 141
Number of TEPs                   : 0
Number of Local EAD Entries      : 178
Number of Remote EAD Entries     : 0
Number of Local MAC Routes       : 89
          MAC                    : 89
          MAC-IPv4               : 0
          MAC-IPv6               : 0
Number of Local ES:Global MAC    : 1
Number of Remote MAC Routes      : 0
          MAC                    : 0
          MAC-IPv4               : 0
          MAC-IPv6               : 0
Number of Remote SYNC MAC Routes : 0
Number of Local IMCAST Routes    : 89
Number of Remote IMCAST Routes   : 0
Number of Internal Labels        : 0
Number of single-home Internal IDs : 0
Number of multi-home Internal IDs  : 0
Number of ES Entries             : 3
Number of Neighbor Entries       : 0
EVPN Router ID                   : 192.168.10.1
BGP ASN                          : 64600
PBB BSA MAC address              : d46a.3599.50d8
Global peering timer             :      3 seconds
Global recovery timer            :     30 seconds
Global carving timer             :      0 seconds
Global MAC postpone timer        :    300 seconds [not running]
Global core de-isolation timer   :    120 seconds [not running]
```

```
EVPN services costed out on node   : No
          Startup-cost-in timer    : Not configured
          EVPN manual cost-out     : No
          EVPN Bundle Convergence  : No
```

Let's bring up the core interfaces and see how the *core de-isolation* timer starts.

```
Router# rollback configuration last 1

Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
6 items committed in 1 sec (5)items/sec
Updating.
Updated Commit database in 1 sec
Configuration successfully rolled back 1 commits.
```

In this example, you can see that the *core de-isolation* timer starts running after the core interfaces come up. When the core interfaces are UP, the state of core changes to Deisolating. In the following output you can see the state as Deisolating and core interfaces are up and the *core de-isolation* timer has started.

The access interfaces come up only after the *core de-isolation* timer value expires. In the following output you can see the access interfaces are still down.

```
Router# show evpn group

EVPN Group: 100

  state: Deisolating

  Core Interfaces:
     Bundle-Ether11: up
     Bundle-Ether21: up

  Access Interfaces:
     Bundle-Ether200: down
     Bundle-Ether201: down

Router# show evpn summary
-----------------------------
Global Information
-----------------------------
Number of EVIs                    : 141
Number of TEPs                    : 2
Number of Local EAD Entries       : 178
Number of Remote EAD Entries      : 534
Number of Local MAC Routes        : 89
          MAC                     : 89
          MAC-IPv4                : 0
          MAC-IPv6                : 0
Number of Local ES:Global MAC     : 1
Number of Remote MAC Routes       : 0
          MAC                     : 0
          MAC-IPv4                : 0
          MAC-IPv6                : 0
Number of Remote SYNC MAC Routes  : 0
Number of Local IMCAST Routes     : 89
Number of Remote IMCAST Routes    : 178
Number of Internal Labels         : 178
Number of single-home Internal IDs : 0
Number of multi-home Internal IDs  : 0
Number of ES Entries              : 3
```

```
Number of Neighbor Entries         : 178
EVPN Router ID                     : 192.168.10.1
BGP ASN                            : 64600
PBB BSA MAC address                : d46a.3599.50d8
Global peering timer               :      3 seconds
Global recovery timer              :     30 seconds
Global carving timer               :      0 seconds
Global MAC postpone timer          :    300 seconds [not running]
Global core de-isolation timer     :    120 seconds [running, 14.6 sec left]
EVPN services costed out on node   : No
          Startup-cost-in timer    : Not configured
          EVPN manual cost-out     : No
          EVPN Bundle Convergence  : No
```

The following output shows that the *core de-isolation* timer has expired.

```
Router# show evpn summary
----------------------------
Global Information
----------------------------
Number of EVIs                     : 141
Number of TEPs                     : 2
Number of Local EAD Entries        : 178
Number of Remote EAD Entries       : 534
Number of Local MAC Routes         : 89
          MAC                      : 89
          MAC-IPv4                 : 0
          MAC-IPv6                 : 0
Number of Local ES:Global MAC      : 1
Number of Remote MAC Routes        : 0
          MAC                      : 0
          MAC-IPv4                 : 0
          MAC-IPv6                 : 0
Number of Remote SYNC MAC Routes   : 0
Number of Local IMCAST Routes      : 89
Number of Remote IMCAST Routes     : 178
Number of Internal Labels          : 178
Number of single-home Internal IDs : 0
Number of multi-home Internal IDs  : 0
Number of ES Entries               : 3
Number of Neighbor Entries         : 178
EVPN Router ID                     : 192.168.10.1
BGP ASN                            : 64600
PBB BSA MAC address                : d46a.3599.50d8
Global peering timer               :      3 seconds
Global recovery timer              :     30 seconds
Global carving timer               :      0 seconds
Global MAC postpone timer          :    300 seconds [not running]
Global core de-isolation timer     :    120 seconds [not running]
EVPN services costed out on node   : No
          Startup-cost-in timer    : Not configured
          EVPN manual cost-out     : No
          EVPN Bundle Convergence  : No
```

After the *core de-isolation* timer expires, you can see that the state is Ready, and both core and access interfaces are UP.

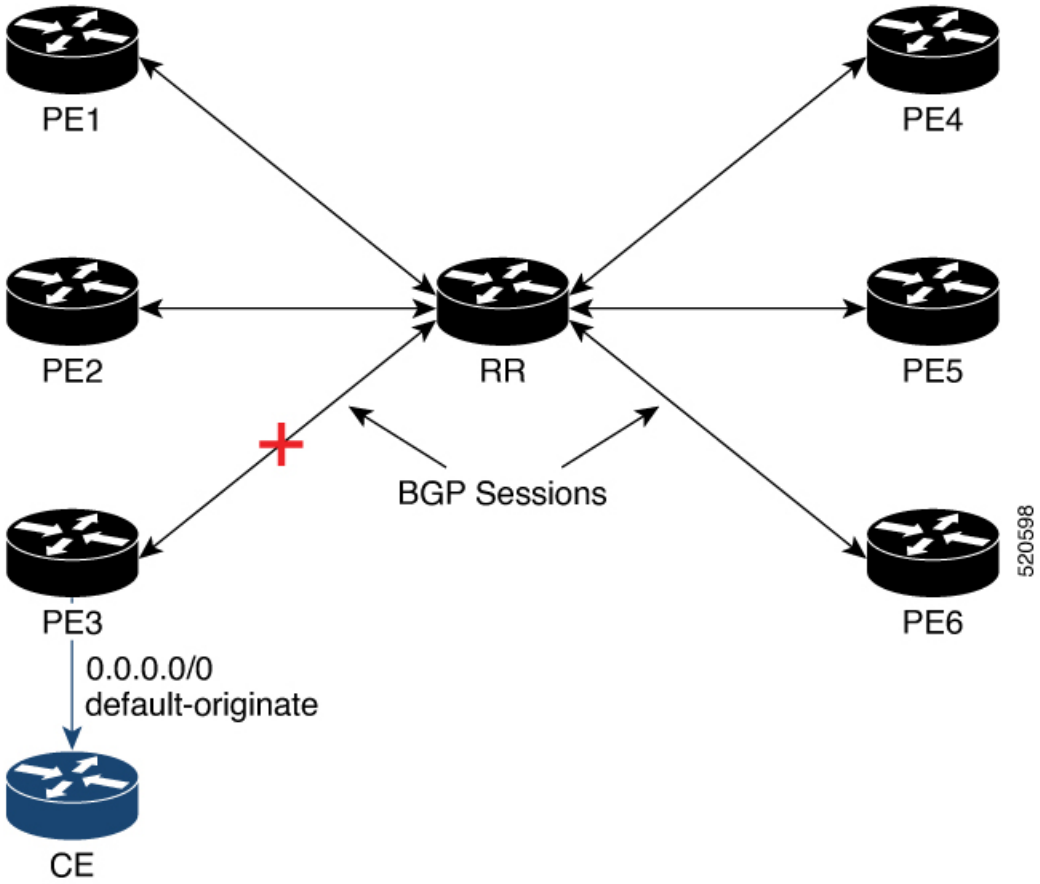
```
Router# show evpn group

EVPN Group: 100

  state: Ready
```

```
Core Interfaces:
   Bundle-Ether11: up
   Bundle-Ether21: up

Access Interfaces:
   Bundle-Ether200: up
   Bundle-Ether201: up
```

Conditional Advertisement of Default-Originate

The router advertises the default-originate (0.0.0.0/0) towards the network fabric only upon receiving all the core routes. The router withdraws the advertisement of default-originate when the core is isolated. To avoid traffic drop, install the routes in the hardware. To accommodate an additional delay for the routes to be installed in the hardware, you can configure a timeout for the installed routes.

Figure 9: Advertisement of default-originate



In this topology, PE3 advertises the default-originate to CE only when the PE3 session to RR is established and all the routes are received from the RR.

Configure Conditional Advertisement of Default-Originate

Perform the following tasks to configure conditional advertisement of default-originate.

- Configure BGP
- Configure RPL
- Track BGP neighbor address-family state

Configuration Example

Perform the following task on PE3:

```
/* Configure BGP */
Router# configure
Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 192.0.2.1
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 172.16.0.5
Router(config-bgp-nbr)# remote-as 200
Router(config-bgp-nbr)# address-family vpnv4 unicast
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# exit
Router(config-bgp)# vrf cust1
Router(config-bgp-vrf)# rd auto
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# redistribute static
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# neighbor 172.16.0.5
Router(config-bgp-vrf-nbr)# remote-as 200
Router(config-bgp-vrf-nbr)# address-family ipv4 unicast
Router(config-bgp-vrf-nbr-af)# default-originate route-policy track-bgp-core-policy
Router(config-bgp-vrf-nbr-af)# route-policy pass in
Router(config-bgp-vrf-nbr-af)# route-policy pass out
Router(config-bgp-vrf-nbr-af) commit


/* Configure RPL */
Router# configure
Router(config)# route-policy track-bgp-core-policy
Router(config-rpl)# if track core-group-1 is up then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
Router(config-rpl)# end-policy
Router(config)# commit


/* Track BGP neighbor address-family state */
Router# configure
Router(config)# track core-group-1
Router(config-track)# type bgp neighbor address-family state
Router(config-track-bgp-nbr-af)# address-family vpnv4 unicast
Router(config-track-bgp-neighbor)# neighbor 172.16.0.5
Router(config-track-bgp-neighbor)# commit
```

Running Configuration

This section shows conditional advertisement of default-originate running configuration.

```
configure
 router bgp 100
```

```
  bgp router-id 192.0.2.1
  address-family vpnv4 unicast
!
 neighbor 172.16.0.5
  remote-as 200
  address-family vpnv4 unicast
!

 vrf cust1
  rd auto
  address-family ipv4 unicast
   redistribute connected
   redistribute static
!
 neighbor 172.16.0.5
  remote-as 200
  address-family ipv4 unicast
   default-originate route-policy track-bgp-core-policy
   route-policy pass in
   route-policy pass out
!

route-policy track-bgp-core-policy
 if track core-group-1 is up then
  pass
  endif
 end-policy
!
track network-core
 type bgp neighbor address-family state
  address-family vpnv4 unicast
   neighbor 172.16.0.5
!
```

Verification

Verify conditional advertisement of default-originate.

```
Router# show rpl active route-policy
Wed May 27 06:54:31.902 UTC

ACTIVE -- Referenced by at least one policy which is attached
INACTIVE -- Only referenced by policies which are not attached
UNUSED -- Not attached (directly or indirectly) and not referenced

The following policies are (ACTIVE)
-----------------------------------------
    track-bgp-core
-----------------------------------------
Router# show rpl route-policy track-bgp-core-policy
Wed May 27 06:54:38.090 UTC
route-policy track-bgp-core-policy
  if track core-group-1 is up then
    pass
  endif
end-policy
!

Router# show bgp policy route-policy track-bgp-core-policy summary
Wed May 27 06:54:42.823 UTC
Network            Next Hop        From            Advertised to
0.0.0.0/0          0.0.0.0         Local           172.16.0.5
```

```
Router# show bgp neighbor 172.16.0.5
Wed May 27 06:55:39.535 UTC

BGP neighbor is 172.16.0.5
 Remote AS 9730, local AS 9730, internal link
 Remote router ID 172.16.0.5
  BGP state = Established, up for 10:41:12
[snip]
 For Address Family: IPv4 Unicast
  BGP neighbor version 2
  Update group: 0.4 Filter-group: 0.1  No Refresh request being processed
  Default information originate: default route-policy track-bgp-core-policy, default sent
  AF-dependent capabilities:
[snip]
  Track Enabled, Status UP, Nbr GR state Not Enabled, EOR tmr Not Running
  Advertise routes with local-label via Unicast SAFI
```

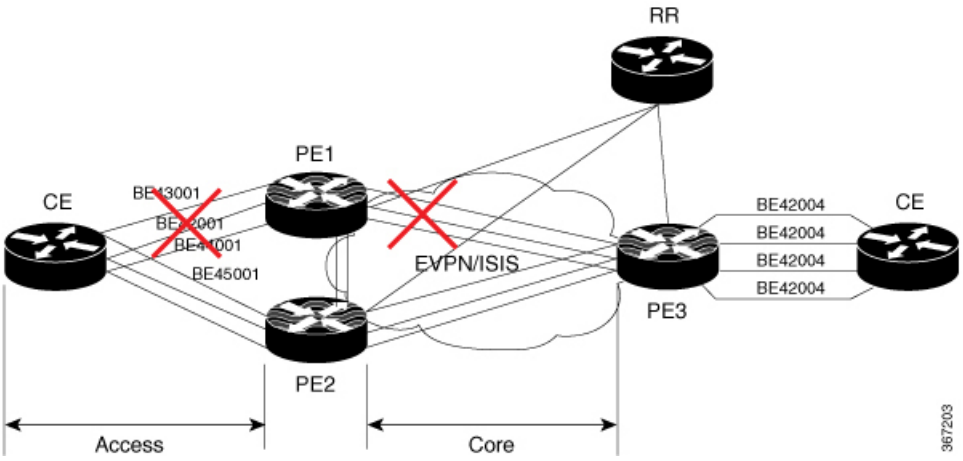
EVPN Core Isolation Protection

The EVPN Core Isolation Protection feature enables you to monitor and detect the link failure in the core. When a core link failure is detected in the provider edge (PE) device, EVPN brings down the PE's Ethernet Segment (ES), which is associated with access interface attached to the customer edge (CE) device.

EVPN replaces ICCP in detecting the core isolation. This new feature eliminates the use of ICCP in the EVPN environment.

Consider a topology where CE is connected to PE1 and PE2. PE1, PE2, and PE3 are running EVPN over the MPLS core network. The core interfaces can be Gigabit Ethernet or bundle interface.

Figure 10: EVPN Core Isolation Protection



When the core links of PE1 go down, the EVPN detects the link failure and isolates PE1 node from the core network by bringing down the access network. This prevents CE from sending any traffic to PE1. Since BGP session also goes down, the BGP invalidates all the routes that were advertised by the failed PE. This causes the remote PE2 and PE3 to update their next-hop path-list and the MAC routes in the L2FIB. PE2 becomes the forwarder for all the traffic, thus isolating PE1 from the core network.

When all the core interfaces and BGP sessions come up, PE1 advertises Ethernet A-D Ethernet Segment (ES-EAD) routes again, triggers the service carving and becomes part of the core network.

Configure EVPN Core Isolation Protection

Configure core interfaces under EVPN group and associate that group to the Ethernet Segment which is an attachment circuit (AC) attached to the CE. When all the core interfaces go down, EVPN brings down the associated access interfaces which prevents the CE device from using those links within their bundles. All interfaces that are part of a group go down, EVPN brings down the bundle and withdraws the ES-EAD route.

Starting from Cisco IOS-XR software version 7.1.2, you can configure a sub-interface as an EVPN Core. With this enhancement, when using IOS-XR software versions 7.1.2 and above, EVPN core facing interfaces can be physical, bundle main, or sub-interfaces. For all Cisco IOS-XR software versions lower than 7.1.2, EVPN core facing interfaces must be physical or bundle main. Sub-interfaces are not supported.

EVPN core facing interfaces can be physical main interface or subinterface, or bundle main interface or subinterface.

Restrictions

- A maximum of 24 groups can be created under the EVPN.
- A maximum of 12 core interfaces can be added under the group.
- The core interfaces can be reused among the groups. The core interface can be a bundle interface.
- EVPN group must only contain core interfaces, do not add access interfaces under the EVPN group.
- The access interface can only be a bundle interface.
- EVPN core facing interfaces must be physical or bundle main interfaces only. Sub-interfaces are not supported.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# group 42001
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/1
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/3
Router(config-evpn-group)#exit
!
Router(config-evpn)# group 43001
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/2
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/4
Router(config-evpn-group)#exit
!
Router# configure
Router(config)# evpn
Router(config-evpn)# interface bundle-Ether 42001
Router(config-evpn-ac)# core-isolation-group 42001
Router(config-evpn-ac)# exit
!
Router(config-evpn)# interface bundle-Ether 43001
Router(config-evpn-ac)# core-isolation-group 43001
Router(config-evpn-ac)# commit
```

Running Configuration

```
configure
 evpn
```

```
  group 42001
   core interface GigabitEthernet0/2/0/1
   core interface GigabitEthernet0/2/0/3
   !
  group 43001
   core interface GigabitEthernet0/2/0/2
   core interface GigabitEthernet0/2/0/4
   !
 !
configure
 evpn
  interface bundle-Ether 42001
   core-isolation-group 42001
   !
  interface bundle-Ether 43001
   core-isolation-group 43001
   !
  !
```

Verification

The **show evpn group** command displays the complete list of evpn groups, their associated core interfaces and access interfaces. The status, up or down, of each interface is displayed. For the access interface to be up, at least one of the core interfaces must be up.

```
Router# show evpn group /* Lists specific group with core-interfaces and access interface
status */
EVPN Group: 42001
  State: Ready
  Core Interfaces:
     Bundle-Ethernet110: down
     Bundle-Ethernet111: down
     GigabethEthernet0/2/0/1: up
     GigabethEthernet0/2/0/3: up
     GigabethEthernet0/4/0/8: up
     GigabethEthernet0/4/0/9: up
     GigabethEthernet0/4/0/10: up
  Access Interfaces:
     Bundle-Ether42001: up

EVPN Group: 43001
  State: Ready
  Core Interfaces:
     Bundle-Ethernet110: down
     GigabethEthernet0/2/0/2: up
     GigabethEthernet0/2/0/4: up
     GigabethEthernet0/4/0/9: up

  Access Interfaces:
     Bundle-Ether43001: up
```

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

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.

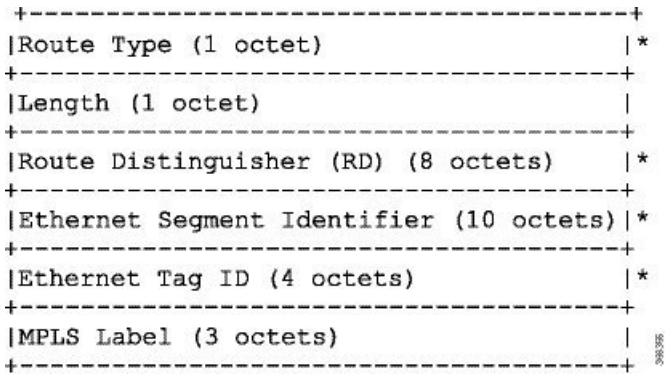
EVPN Route Types

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:



NLRI Format: Route-type 1:

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

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

Path attributes: `[MPLS Label]`

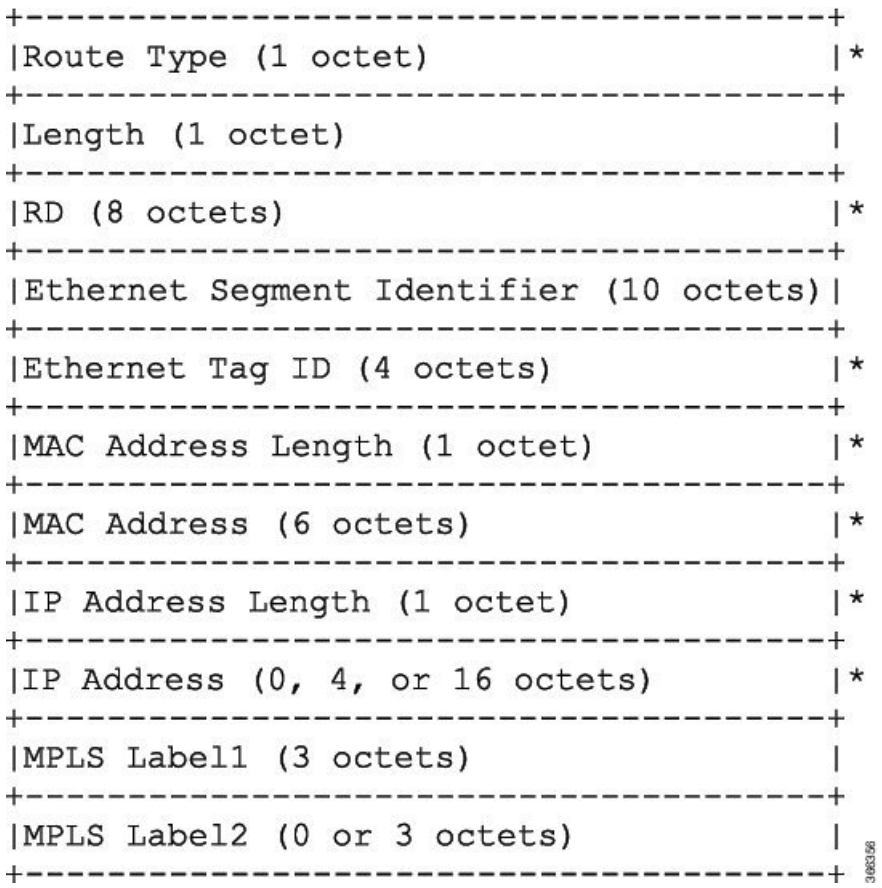
Example

```
route-policy evpn-policy
  if rd in (10.0.0.1:0) [and/or evpn-route-type is 1] [and/or esi in
(0a1.a2a3.a4a5.a6a7.a8a9)] [and/or etag is 4294967295] then
    set ..
  endif
end-policy
!
route-policy evpn-policy
  if rd  in (1.0.0.2:0) [and/or evpn-route-type is 1] [and/or esi in
(00a1.a2a3.a4a5.a6a7.a8a9)] [and/or etag is 4294967295] then
    set ..
  endif
end-policy
```

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:



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]`

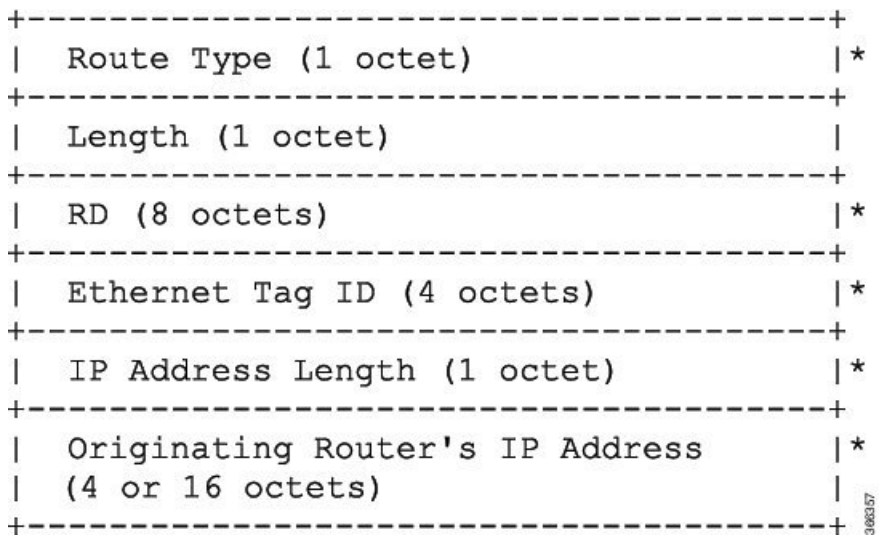
Example

```
route-policy evpn-policy
  if rd in (10.0.0.2:0) [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:



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]`

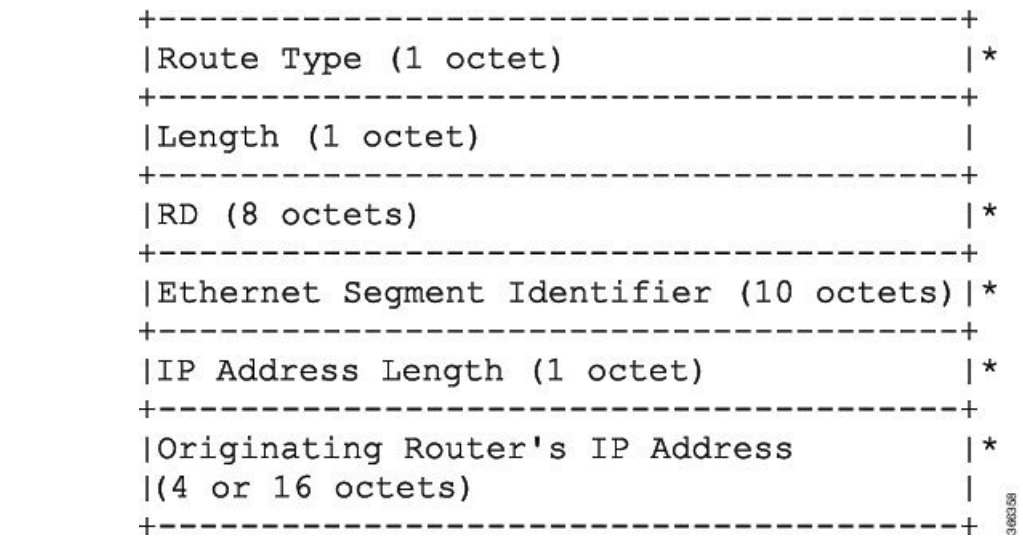
Example

```
route-policy evpn-policy
  if rd  in (10.0.0.1:300) [and/or evpn-route-type is 3] [and/or etag is 0] [and/or
evpn-originator in (10.0.0.1)] 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:



NLRI Format: Route-type 4:

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

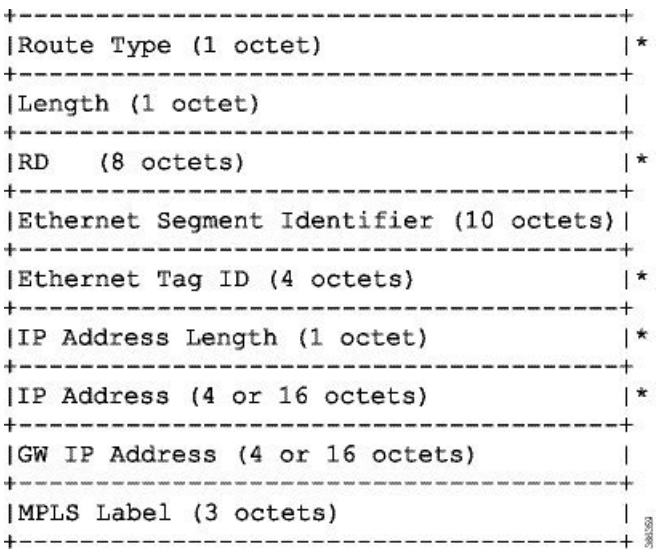
Net attributes: `[Type][RD][ESI][IP Addr Len][Originating Router's IP Addr]`

Example

```
route-policy evpn-policy
  if rd  in (10.0.0.1:0) [and/or evpn-route-type is 4] [and/or esi in
(00a1.a2a3.a4a5.a6a7.a8a9)] [and/or evpn-originator in (10.0.0.1)] then
    set ..
  endif
end-policy
```

Route Type 5: IP Prefix Route

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



NLRI Format: Route-type 5:

```
[Type][Len][RD][ESI][ETag][IP Addr Len][IP Addr][GW IP Addr][Label]
```

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

Path attributes: `[ESI], [GW IP Addr], [Label]`

Example

```
route-policy evpn-policy
  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
```

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

```
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

```
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

```
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

```
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.

```
/* 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)# !
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 l2vpn evpn
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 l2vpn evpn
Router(config-bgp-nbr-af)# route-policy policy_use_pass_mac_set in
Router(config-bgp-nbr-af)# commit

/* Configuring a esi-set and refering 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 refering 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
end-policy
!
router bgp 100
 address-family l2vpn evpn
 !
 neighbor 10.0.0.10
  remote-as 8
  address-family l2vpn evpn
  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
end-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
end-policy
!
route-policy ex_3
  if evpn-route-type is 5 then
    set extcommunity bandwidth (100:9999)
  else
    pass
  endif
end-policy
!
route-policy samp
end-policy
!
route-policy samp1
  if rd in (30.0.101.2:0) then
    pass
  endif
end-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
!
```

CFM on EVPN ELAN

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
CFM on EVPN ELAN	Release 7.4.1	This feature allows you to effectively manage a network with EVPN services running EVPN ELAN and helps you to monitor the ELAN services, thereby providing high-speed Layer 2 and Layer 3 services with high resiliency. This feature is now supported on routers that have Cisco NC57 line cards installed and operate in native mode only. The following offload types are supported: • Hardware (HW) Offload • Non-Offload • Software (SW) Offload

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
CFM on EVPN ELAN	Release 7.6.1	Connectivity fault management (CFM) enables monitoring an Ethernet network with multiple service instances. With CFM now supporting single-homed EVPN Emulated Local Area Network (ELAN) services, you can monitor users' services against their contractual service level agreements. This removes the operational complexity of managing different market segments and subscribers on your network.

Connectivity fault management (CFM) is a service-level Operations and Maintenance (OAM) protocol that provides tools for monitoring and troubleshooting end-to-end Ethernet services for each VLAN. This includes proactive connectivity monitoring, fault verification, and fault isolation.

Supported Offload Types and Timer Values

The following are supported offload types:

- Hardware (HW) Offload type: The check message (CCM) timers for a CFM session on a physical interface is less than one second.

Note The Hardware (HW) Offload type is supported only in Cisco NC 57 line cards.

- Non-Offload type: The CCM timers for a CFM session on a physical interface is greater than one second.
- Software (SW) Offload type: The CFM session on a bundle interface.

The following are the supported timer values:

- 10s: Interval of 10 seconds
- 1m: Interval of 1 minute
- 10m: Interval of 10 minutes
- Non-Offload type supports 10s, 1m on the physical interface
- SW Offload type supports 10s,1m, LAG 3.33
- CCM interval of 10m is not supported on NCS57 line cards operating in native mode.
- A maximum of 8K CFM UP MEP sessions are supported because EVPN ELAN supports 8K bridge domains on NCS57 line cards operating in native mode.
- You can configure both UP and DOWN MEPs on the same L2VPN with EVPN ELAN in the Cisco NC 57 line cards.
- Supports 3.3ms,10ms,100ms,1s CCM timers for HW-Offload UP MEP with EVPN ELAN on NCS57 line cards operating in native mode.

Restrictions for CFM on EVPN ELAN

CFM on EVPN ELAN is subjected to these restrictions:

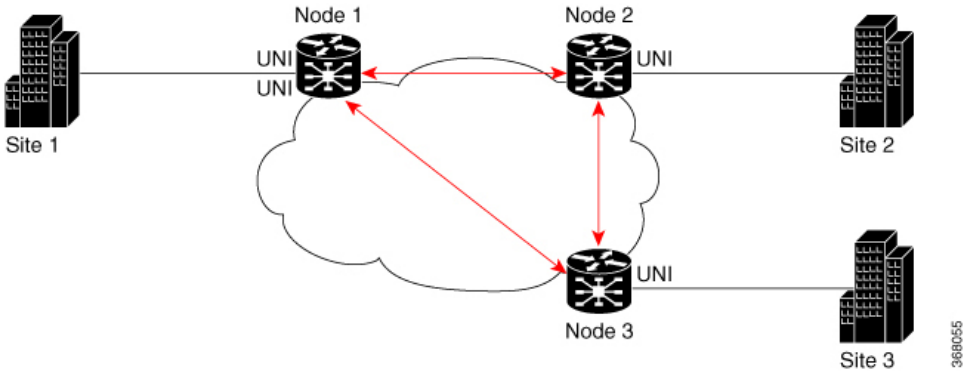
- Supports only single-homed EVPN ELAN.
- Supports single homing with one AC per PW.
- DOWN MEP on AC interface of EVPN-BD is not supported.
- Does not support loss measurement.
- Does not support Y1731.
- CFM over EVPN ELAN with MEPs along with multiple AC scenarios supports CCM and does not support LBM or LBR.

CFM on EVPN ELAN does not support the following configurations:

- UP MEP of different domain and same level on same EVPN-BD
- UP MEP of different level on different AC part of same BD as all AC interfaces are part of same service provider domain (EVPN-BD) in PE.

Configure CFM on EVPN ELAN

Figure 11: CFM on EVPN ELAN: Full Mesh Topology



Node 1, 2 and 3 in this topology can be Cisco routers.

Configuring CFM on EVPN ELAN involves these main tasks:

- Enabling CFM service continuity check
- Configuring MEP cross-check
- Enabling CFM for the interface

Configuration Example for CFM on EVPN ELAN: Full Mesh Topology

```
/* Enabling CFM continuity check */
Router# ethernet cfm
Router(config-cfm# domain bd-domain level 1 id null
Router(config-cfm-dmn)# service bd-domain bridge group bg-elan bridge-domain bd-elan id
icc-based MC MCMC
Router(config-cfm-dmn-svc)# continuity-check interval 1m
/* Configuring MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-dmn-svc)# mep-id 1112
Router(config-cfm-dmn-svc)# mep-id 1113
Router(config-cfm-dmn-svc)# commit
```

Repeat the above configurations for node 2 and node 3, with the respective mep-id values. For node 2, configure MEP cross-check with respective mep-id values of node 1 and node 3 (1111 and 1113 respectively, in this example). For node 3, configure MEP cross-check with respective mep-id values of node 1 and node 2 (1111 and 1112 respectively, in this example).

```
/* Enabling CFM on the interface */
Router(config)# interface gigabitEthernet 0/0/0/0.100 l2transport
Router(config-subif)# description bg-elan
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# rewrite ingress tag pop 1 symmetric
Router(config-subif)# mtu 9100
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 1111
Router(config-if-cfm-mep)# commit
```

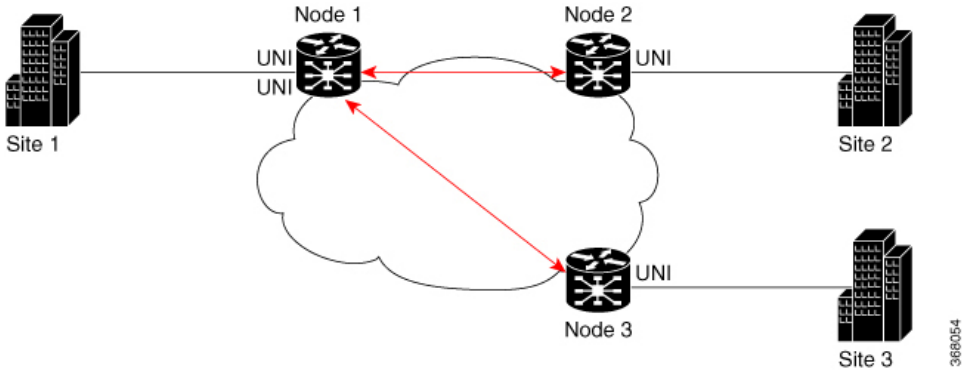
You must repeat the above configurations for node 2 and node 3, with the respective *mep-id* values (that is, 1112 for node 2 and 1113 for node 3, in this example).

Running Configuration for CFM on EVPN ELAN: Full Mesh Topology

This sections shows the running configuration on node 1.

```
ethernet cfm
 domain bd-domain level 1 id null
  service bd-domain bridge group bg-elan bridge-domain bd-elan id icc-based MC MCMC
   continuity-check interval 1m
   mep crosscheck
    mep-id 1112
    mep-id 1113
   !
  !
 !
!

interface GigabitEthernet0/0/0/0.100 l2transport
 description bg-elan
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 mtu 9100
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 1111
 !
```

Figure 12: CFM on EVPN ELAN: Hub and Spoke Topology



Configuration Example for CFM on EVPN ELAN: Hub and Spoke Topology

The CFM configuration for the hub and spoke topology remains the same as that of full mesh topology mentioned above, except for these additional steps for SLA profile configuration to be done under the interface.

```
/* 1112 and 1113 in this example, are the mep-id values of node 2 and node 3 */
Router(config)#interface gigabitEthernet 0/0/0/0.100 l2transport
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 1111
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1112
Router(config-if-cfm-mep)# sla operation profile test-profile2 target mep-id 1112
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1113
Router(config-if-cfm-mep)# sla operation profile test-profile2 target mep-id 1113
Router(config-if-cfm-mep)# commit
```

Running Configuration for CFM on EVPN ELAN: Hub and Spoke Topology

This sections shows the running configuration on node 1.

```
interface GigabitEthernet0/0/0/0.100 l2transport
 description bg-elan
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 mtu 9100
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 1111
   sla operation profile test-profile1 target mep-id 1112
   sla operation profile test-profile2 target mep-id 1112
   sla operation profile test-profile1 target mep-id 1113
   sla operation profile test-profile2 target mep-id 1113
 !
```

Example 1: The below configurations details UP MEPs of same domain and level on the following:

- Multiple AC per BD on local
- Single AC per BD on remote

```
Router#sh run ethernet cfm
Tue Sep 14 19:08:32.666 UTC
ethernet cfm
 domain BD-DOMAIN level 4 id null
  service BD-SERVICE bridge group ELAN_FUNC_3 bridge-domain FUNC_3 id number 100
   continuity-check interval 10s
   mep crosscheck
    mep-id 5
    mep-id 1101
    mep-id 1103


Router#sh run l2vpn
Tue Sep 14 19:08:37.979 UTC
l2vpn
 bridge group ELAN_FUNC_3
  bridge-domain FUNC_3
   interface TenGigE0/0/0/0.1
   !
   interface TenGigE0/0/0/1.2
   !
   evi 101

Router#sh run int Te0/0/0/0.1
Tue Sep 14 19:08:42.677 UTC
interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 1103

Router#sh run int Te0/0/0/1.2
Tue Sep 14 19:08:49.485 UTC
interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 5
```

Example 2: The below configurations details multiple UP MEPs of same domain and level on AC interfaces that are part of the same BD:

```
Router#sh run ethernet cfm
ethernet cfm
 domain BD-DOMAIN level 4 id null
  service BD-SERVICE bridge group ELAN_FUNC_3 bridge-domain FUNC_3
   continuity-check interval 10s
   mep crosscheck
    mep-id 1
    mep-id 2
    mep-id 21
    mep-id 22
domain BD-DOMAIN1 level 3 id null
  service BD-SERVICE1 bridge group ELAN_FUNC_3 bridge-domain FUNC_3
   continuity-check interval 10s
   mep crosscheck
    mep-id 1001
    mep-id 1021
    mep-id 2001
    mep-id 2021


Router#sh run int Te0/0/0/0
 interface TenGigE0/0/0/0
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 21
  mep domain BD-DOMAIN1 service BD-SERVICE1 mep-id 1021
 l2transport


Router#sh run int Te0/0/0/1
interface TenGigE0/0/0/1
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 22
  mep domain BD-DOMAIN1 service BD-SERVICE1 mep-id 2021
 l2transport


Router#sh run l2vpn
l2vpn
 bridge group ELAN_FUNC_3
  bridge-domain FUNC_3
   interface TenGigE0/0/0/0
   interface TenGigE0/0/0/1
   Interface TenGigE0/0/0/2
   evi 101
```

Example 3: The below configurations details multiple services for different EVPN-BD on same domain level :

```
Router#sh run ethernet cfm
Tue Sep 14 19:22:01.196 UTC
ethernet cfm
 domain evpn-bd level 4 id null
  service evpn-bd1 bridge group BG1 bridge-domain BD1
   continuity-check interval 10s
   mep crosscheck
    mep-id 5
    mep-id 6
```

```
    mep-id 1101
    mep-id 1103
  service evpn-bd2 bridge group BG2 bridge-domain BD2
   continuity-check interval 10s
   mep crosscheck
    mep-id 11
    mep-id 21
    mep-id 101

Router#sh run l2vpn
l2vpn
 bridge group BG1
  bridge-domain BD1
   interface TenGigE0/0/0/0.1
   interface TenGigE0/0/0/1.2
   evi 101
bridge group BG2
  bridge-domain BD2
   interface TenGigE0/0/0/2.1
   interface TenGigE0/0/0/5.2
   evi 201

Router#sh run int Te0/0/0/0.1
Tue Sep 14 19:22:12.368 UTC
interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 1103

Router#sh run int Te0/0/0/1.2
Tue Sep 14 19:22:19s.258 UTC
interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 5

Router#sh run int Te0/0/0/2.1
Tue Sep 14 19:22:23.539 UTC
interface TenGigE0/0/0/2.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd service evpn-bd2 mep-id 101

Router#sh run int Te0/0/0/5.2
Tue Sep 14 19:22:27.954 UTC
interface TenGigE0/0/0/5.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd service evpn-bd2 mep-id 11
```

Example 4: The below configurations details different EVPN-BD on different domain levels:

```
Router#sh run ethernet cfm
Tue Sep 14 19:39:39.522 UTC
ethernet cfm
 domain evpn-bd level 4 id null
  service evpn-bd1 bridge group BG1 bridge-domain BD1
   continuity-check interval 10s
   mep crosscheck
    mep-id 5
    mep-id 6
    mep-id 1101
```

```
    mep-id 1103
   !
  !
 !
 domain evpn-bd2 level 3 id null
  service evpn-bd2 bridge group BG2 bridge-domain BD2
   continuity-check interval 10s
   mep crosscheck
    mep-id 11
    mep-id 21
    mep-id 101
    mep-id 201
   !
  !
 !
!
Router#sh run l2vpn
Tue Sep 14 19:39:44.004 UTC
l2vpn
 bridge group BG1
  bridge-domain BD1
   interface TenGigE0/0/0/0.1
   !
   interface TenGigE0/0/0/1.2
   !
   evi 101
   !
  !
 !
 bridge group BG2
  bridge-domain BD2
   interface TenGigE0/0/0/2.1
   !
   interface TenGigE0/0/0/5.2
   !
   evi 201
   !
  !
 !
!
Router#sh run int Te0/0/0/0.1
Tue Sep 14 19:39:50.042 UTC
interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 1103
  !
 !
!

Router#sh run int Te0/0/0/1.2
Tue Sep 14 19:39:53.798 UTC
interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 5
  !
 !
!

Router#sh run int Te0/0/0/2.1
Tue Sep 14 19:39:59.176 UTC
interface TenGigE0/0/0/2.1 l2transport
 encapsulation dot1q 1
```

```
 ethernet cfm
  mep domain evpn-bd2 service evpn-bd2 mep-id 101
  !
 !
!

Router#sh run int Te0/0/0/5.2
Tue Sep 14 19:40:03.689 UTC
interface TenGigE0/0/0/5.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd2 service evpn-bd2 mep-id 11
  !
 !
!
```

Related Topics

CFM on EVPN ELAN, on page 86

Associated Commands

- continuity-check
- ethernet cfm
- mep crosscheck
- mep domain
- sla operation

EVPN Bridging and VPWS Services over BGP-LU Underlay

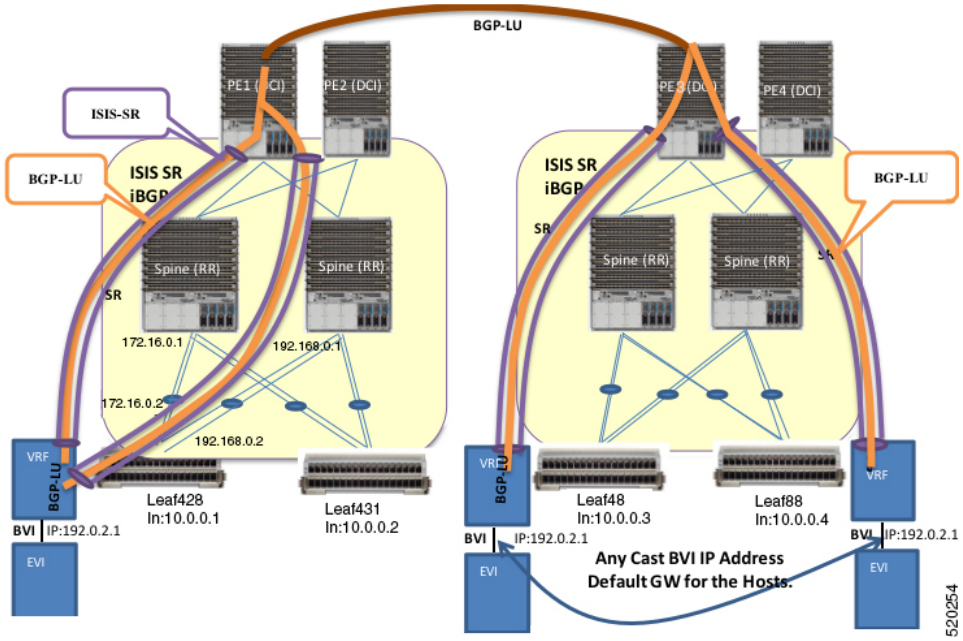
The EVPN Bridging and VPWS Services over BGP-LU Underlay feature allows you to configure end-to-end EVPN services between data centers (DCs). This feature allows you to perform ECMP at three-levels: transport, BGP- LU, and service level.

This feature supports the following services:

- IRB VRF over BGP-LU using IGP (SR or non-SR (LDP or IGP))
- EVPN Aliasing over BGP-LU using IGP (SR or non-SR (LDP or IGP))
- VPWS over BGP-LU using IGP

Note EVPN IRB with default-vrf over BGP-LU over IGP is not supported on the Cisco NCS routers.

Figure 13: EVPN Bridging and VPWS Services over BGP-LU Underlay



This section explains the topology of EVPN Bridging and VPWS Services over BGP-LU Underlay feature:

- Consider two data centers that are connected through DCI. Configure EVPN with bridging and inter-subnet routing on the leaf nodes.
- Configure EVPN instance with BVI attachment circuit to interface with L3-VRF.
- Configure BVI interface with anycast IP address with the same MAC address. This is the default gateway for all the hosts across the same EVPN bridged domain.
- The leaf acts as default gateway for its local hosts.
- Connect hosts to leaf nodes. Leaf nodes are routed across the spines. For DC interconnectivity, the spines are connected through provider edge (PE) device and Data Center Interconnect (DCI).
- IS-IS labelled IGP and I-BGP are enabled internally across the leaf nodes, spine and DCI. The spine acts as a Route Reflector (RR).
- Configure IS-IS SR policy across the leaf node, spine and DCI.
- Configure BGP-LU between the DCs.
- Labelled Unicast BGP routers are learnt across the leaf nodes and tunnelled through IGP labelled paths (IS-IS SR).

  For example, at Leaf428, BGP-LU routes are learnt for remote loopback 10.0.0.3 and 10.0.0.4.
- IRB (BVI) interface routes are learnt across the EVPN instances and programmed as labelled routes tunnelled through BGP-LU.

  For example, at Leaf428, 192.0.2.1 can be reached with two BGP-LU paths 10.0.0.3 and 10.0.0.4.

After establishing the BGP-LU services, you can configure either EVPN instance or EVPN VPWS to support BGP-LU.

Configure EVPN Bridging and VPWS Services over BGP-LU Underlay

Perform these tasks to configure the EVPN Bridging and VPWS Services over BGP-LU Underlay feature.

- Configure IGP
- Configure BGP
- Configure EVPN instance and ESI
- Configure BVI (IRB) Interface
- Configure VRF
- Configure BVI with VRF
- Configure VRF under BGP
- Configure bridge domain and associate with attachment circuits and EVPN instance
- Configure bridge domain and associate with attachment circuits, EVPN instance and BVI
- Configure EVPN VPWS

Configuration Example

```
/* Configure IGP */
IGP configuration is a pre-requisite to configure EVPN. IGP can be OSPF or ISIS.
Router# configure
Router(config)#router ospf 1
Router(config-ospf)#router-id 209.165.201.1
Router(config-ospf)#area 10
Router(config-ospf-ar)#interface loopback0\
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface TenGigE0/0/0/1\
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface TenGigE0/0/0/17\
Router(config-ospf-ar-if)#commit

/* Configure BGP enable BGP-LU, and configure iBGP peering */
Router# configure
Router(config)#router bgp 100
Router(config-bgp)#router-id 209.165.201.1
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#network 209.165.200.225/27
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor-group spines
Router(config-bgp-nbrgrp)#remote-as 100
Router(config-bgp-nbrgrp)#update-source loopback0
Router(config-bgp-nbrgrp)#address-family ipv4 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family ipv6 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family l2vpn evpn
```

```
Router(config-bgp-nbrgrp-af)#advertise vpnv4 unicast re-originated
Router(config-bgp-nbrgrp-af)#advertise vpnv6 unicast re-originated
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)exit
Router(config-bgp)neighbor 209.165.200.225
Router(config-bgp-nbr)#use neighbor-group spines
Router(config-bgp-nbr)#commit

/* Configure VPN4 address-family to enforce iBGP policy. */
Router(config)#router bgp 100
Router(config-bgp)#router-id 209.165.201.1
Router(config-bgp)#ibgp policy out enforce-modifications
Router(config-bgp)#address-family vpnv4 unicast
Router(config-bgp-af)#commit

The above steps establish BGP-LU. You can configure either EVPN VPLS or EVPN VPWS on the
PE routers to support BGP-LU.
/* Configure EVPN instance and ESI */
Router#configure
Router(config)#evpn
Router(config-evpn)#evi 100
Router(config-evpn-instance)#advertise-mac
Router(config-evpn-instance-mac)#exit
Router(config-evpn-instance)#exit
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment identifier type 0 aa.aa.aa.aa.aa.aa.aa.aa.ac
Router(config-evpn-ac-es)#bgp route-target 0011.0011.0012
Router(config-evpn-ac)#commit

/* Configure BVI (IRB) Interface */
Router#configure
Router(config)#interface BVI200
Router(config-if)#ipv4 address 192.0.2.1 255.255.255.0
Router(config-if)#commit

/*  Configure VRF */
Router# configure
Router(config)# vrf vpn2
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 81:2
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target 81:2
Router(config-vrf-af)# commit

/* Configure BVI with VRF */
Router(config)# interface BVI200
Router(config-if)# host-routing
Router(config-if)# vrf vpn72
Router(config-if-vrf)# ipv4 address ipv4 address 192.0.2.1 255.255.255.0
Router(config-if-vrf)# mac-address 10.1111.1
Router(config-if)# commit

/* Configure VRF under BGP */
Router(config)# router bgp 100
Router(config-bgp)# vrf vpn2
Router(config-bgp-vrf)# rd 102:2
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# maximum-paths ibgp 8
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# address-family ipv6 unicast
```

```
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# maximum-paths ibgp 8
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# commit

/* Configure bridge domain and associate with attachment circuits and EVPN instance */
Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg1
Router(config-l2vpn-bg)#bridge-domain bd1
Router(config-l2vpn-bg-bd)#interface BundleEther1.100
Router(config-l2vpn-bg-bd-ac)#evi 100
Router(config-l2vpn-bg-bd-evi)#commit

/* Configure bridge domain and associate with attachment circuits, EVPN instance and BVI
*/
Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg2
Router(config-l2vpn-bg)#bridge-domain bd2
Router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/38.200
Router(config-l2vpn-bg-bd-ac)#routed interface BVI200
Router(config-l2vpn-bg-bd-bvi)#evi 200
Router(config-l2vpn-bg-bd-bvi)#commit
Router(config-l2vpn-bg-bd-bvi)#exit

Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg3
Router(config-l2vpn-bg)#bridge-domain bd3
Router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/38.202
Router(config-l2vpn-bg-bd-ac)#routed interface BVI202
Router(config-l2vpn-bg-bd-bvi)#evi 202
Router(config-l2vpn-bg-bd-bvi)#commit

/* Configure EVPN VPWS */
Router#configure
Router(config)#router bgp 100
Router(config-bgp)#neighbor-group spines
Router(config-bgp-nbrgrp)#remote-as 100
Router(config-bgp-nbrgrp)#update-source loopback0
Router(config-bgp-nbrgrp)#address-family ipv4 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family ipv6 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family l2vpn evpn
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)exit
Router(config-bgp)neighbor 209.165.200.225
Router(config-bgp-nbr)#use neighbor-group spines
Router(config-bgp-nbr)#commit
Router(config-bgp-af)#exit
Router(config-bgp)#exit
Router(config)#l2vpn
Router(config-l2vpn)#xconnect group aa-evpn-vpws
Router(config-l2vpn-xc)#p2p vpws_513
Router(config-l2vpn-xc-p2p)#interface Bundle-Ether1.513
Router(config-l2vpn-xc-p2p)#neighbor evpn evi 513 target 513 source 513
Router(config-l2vpn-xc-p2p)# commit
```

Running Configuration

This section shows flooding disable running configuration.

```
/* Configure IGP */
router ospf 1
 router-id 209.165.201.1
 area 10
  interface Loopback0
  !
  interface TenGigE0/0/0/1
  !
  interface TenGigE0/0/0/17
  !
 !
/* Configure BGP, BGP-LU, and iBGP peering */
router bgp 100
 router-id 209.165.201.1
 bgp graceful-restart
 address-family ipv4 unicast
  redistribute connected
  network 209.165.200.225/27
  allocate-label all
address-family ipv6 unicast
  allocate-label all
neighbor-group spines
  remote-as 100
  update-source loopback0
  address-family ipv4 labeled-unicast multipath
  !
  address-family ipv6 labeled-unicast multipath
  !
  address-family l2vpn evpn
   advertise vpnv4 unicast re-originated
   advertise vpnv6 unicast re-originated
 !
 neighbor 209.165.200.225
  use neighbor-group spines
!

/* Configure VPN4 address-family */
router bgp 100
 router-id 209.165.201.1
 ibgp policy out enforce-modifications
 address-family vpnv4 unicast
!

/* Configure EVPN instance and ESI */
evpn
 evi 100
  advertise-mac
 !
 interface Bundle-Ether1
  ethernet-segment
   identifier type 0 aa.aa.aa.aa.aa.aa.aa.aa.ac
   bgp route-target 0011.0011.0012
  !
 !
!

/* Configuring BVI (IRB) Interface */
configure
 interface BVI200
  ipv4 address 192.0.2.1 255.255.255.0

/* Configure VRF */
vrf vpn2
```

```
 address-family ipv4 unicast
  import route-target 81:2
  !
  !
!
 address-family ipv6 unicast
  import route-target 81:2
  !
  !
!

/* Configure BVI with VRF */
interface BVI200
 host-routing
 vrf vpn72
  ipv4 address ipv4 address ipv4 address 192.0.2.1 255.255.255.0
   mac-address 10.1111.1
!

/* Configure VRF under BGP */
router bgp 100
 vrf vpn2
  rd 102:2
  address-family ipv4 unicast
   label mode per-vrf
   maximum-paths ibgp 8
   redistribute connected
  !
  address-family ipv6 unicast
   label mode per-vrf
   maximum-paths ibgp 8
   redistribute connected
  !
!

/* Configure bridge domain and associate with attachment circuits and EVPN instance */
l2vpn
 bridge group bg1
  bridge-domain b1
   interface Bundle-Ether1.100
   !
   evi 100

/*
bridge group bg2
  bridge-domain bd2
   interface TenGigE0/0/0/38.200
   !
   routed interface BVI200
   !
   evi 200
   !
  !

/* Configurige bridge domain and associate with attachment circuits, EVPN instance and BVI
 */
bridge group bg3
  bridge-domain bd3
   interface TenGigE0/0/0/38.202
   !
   routed interface BVI202
   !
   evi 202
   !
```

```
  !
 !

/* Configure EVPN VPWS */
configure
 router bgp 100
  neighbor-group spines
  remote-as 100
  update-source Loopback0
  address-family ipv4 labeled-unicast multipath
  !
  address-family ipv6 labeled-unicast multipath
  !
  address-family l2vpn evpn

neighbor 209.165.200.225
   use neighbor-group spines
 !
!
 l2vpn
  xconnect group aa-evpn-vpws
   p2p vpws_513
    interface Bundle-Ether1.513
    neighbor evpn evi 513 target 513 source 513
```

Verification

Verify that you have configured EVPN Bridging and VPWS Services over BGP-LU Underlay feature successfully.

Note Load Balancing is not supported for EVPN Bridging over BGP-LU with Multipaths.

```
Router#show cef vrf AIM9 10.0.0.1
Tue Jan 20 22:00:56.233 UTC
10.0.0.1/8, version 4, internal 0x5000001 0x0 (ptr 0x97d34b44) [1], 0x0 (0x0), 0x208
(0x98bef0f0)
 Updated Mar 18 06:01:46.175
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 10.0.0.3/8, 7 dependencies, recursive, bgp-multipath [flags 0x6080]
    path-idx 0 NHID 0x0 [0x972c6f08 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.0.0.3/8 via 16448/0/21
     next hop 192.0.2.1/24 BE128        labels imposed {16111 64013 80002}
   via 100.0.0.88/32, 7 dependencies, recursive, bgp-multipath [flags 0x6080]
    path-idx 1 NHID 0x0 [0x972c6d68 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.0.0.4/8 via 16488/0/21
     next hop 192.0.2.1/24 BE128        labels imposed {16111 64009 80002}

Router#show l2vpn xconnect group aa-evpn-vpws xc-name vpws_513 detail
Wed Jan 22 13:14:05.878 GMT+4

Group aa-evpn-vpws, XC vpws_513, state is up; Interworking none
  AC: Bundle-Ether1.513, state is up
    Type VLAN; Num Ranges: 1
    Rewrite Tags: []
```

```
    VLAN ranges: [513, 513]
    MTU 1500; XC ID 0xa00005f7; interworking none
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
      drops: illegal VLAN 0, illegal length 0
  EVPN: neighbor 24000, PW ID: evi 513, ac-id 513, state is up ( established )
    XC ID 0xc0000001
    Encapsulation MPLS
    Source address 209.165.200.225
    Encap type Ethernet, control word enabled
    Sequencing not set
    LSP : Up
EVPN          Local      Remote
-----------------------------------
Label         29045      1048577
MTU           1500       1500
Control word  enabled    enabled
AC ID         513        513
EVPN type    Ethernet    Ethernet
-----------------------------------

Router# show evpn internal-label vpn-id 513 detail
Tue Jan 28 13:22:19.110 GMT+4

VPN-ID  Encap Ethernet Segment Id          EtherTag  Label
-----------------------------------------------------------------
513      MPLS 0099.9900.0000.0000.9999        0      None
 Multi-paths resolved: FALSE (Remote all-active)
 Multi-paths Internal label: None
  EAD/ES  10.0.0.5                           0

513 MPLS   0099.9900.0000.0000.9999          513    24000
 Multi-paths resolved: TRUE (Remote all-active)
 Multi-paths Internal label: 24000
 EAD/ES 10.0.0.5                            0
 EAD/EVI (P) 10.0.0.5                      29104
 Summary pathlist:
0xffffffff (P) 10.0.0.5                    29104
-----------------------------------------------------------------

Router# show mpls forwarding labels 24000 hardware egress detail location 0/0/CPU0

Tue Jan 28 13:22:19.110 GMT+4
Label  Label       or ID              Interface       Switched
-----------------------------------------------------------------
24000  29104       EVPN:513            10.0.0.5       N/A
     Updated: Oct 18 13:14:02.193
     Version: 137839, Priority: 3
     Label Stack (Top -> Bottom): { 29104 }
     NHID: 0x0, Encap-ID: 0x140ea00000002, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 0/4, MTU: 0
     Packets Switched: 0



 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
    PI:0x308de88fb8 PD:0x308de89058 rev:5554240 type: MPLS (2)
    LEAF location: LEM
    FEC key: 0x23e0220000d71
```

```
    label action: MPLS_NOP
LWLDI:
       PI:0x309faa82c8 PD:0x309faa8308 rev:5554239 p-rev:5459825 5459825  ldi type:EOS0_EOS1

        FEC key: 0x23e0220000d71 fec index: 0x0(0) num paths:2, bkup paths: 0
        Collpased IMP LDI: ECD_MARKED
        IMP pattern:3
        PI:0x309faa82c8 PD:0x309faa8308 rev:5554239 p-rev:5459825 5459825
        FEC key: 0x257c720000d71 fec index: 0x20000003(3) num paths:2
        Path:0  fec index: 0x20018f14(102164) DSP fec index: 0x200001f8(504),
                MPLS encap key: 0xf1b00000400140ea MPLS encap id: 0x400140ea Remote: 0
                Label Stack: 29104 16012  dpa-rev:55458217
        Path:1  fec index: 0x20018f15(102165) DSP fec index: 0x200001f9(505),
                MPLS encap key: 0xf1b00000400140eb MPLS encap id: 0x400140eb Remote: 0
                Label Stack: 29104 16012  dpa-rev:55458218

 REC-SHLDI HAL PD context :
ecd_marked:10, collapse_bwalk_required:0, load_shared_lb:0

    RSHLDI:
        PI:0x3093d16af8 PD:0x3093d16bc8 rev:5494421 dpa-rev:36033167 flag:0x1
        FEC key: 0x249e440000d71 fec index: 0x2001c169(115049) num paths: 1
        p-rev:5459825
        Path:0 fec index: 0x2001c169(115049) DSP fec index: 0x200001f8(504),

 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:1, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0,

HW Walk:
LEAF:
    PI:0x308de433b8 PD:0x308de43458 rev:5459864 type: MPLS (2)
    LEAF location: LEM
    FEC key: 0

    LWLDI:
        PI:0x309ffe9798 PD:0x309ffe97d8 rev:5459825 p-rev:4927729 4927729  ldi
type:IMP_EOS0_EOS1
        FEC key: 0x1a1c740000d71 fec index: 0x0(0) num paths:2, bkup paths: 0
        IMP LDI: ECD_MARKED  SERVICE_MARKED
        IMP pattern:3
        PI:0x309ffe9798 PD:0x309ffe97d8 rev:5459825 p-rev:4927729 4927729
        FEC key: 0x23e0220000d71 fec index: 0x20000002(2) num paths:2
        Path:0  fec index: 0x2001f8b4(129204) DSP fec index: 0x200001f8(504),
                MPLS encap key: 0xf1b0000040013ef0 MPLS encap id: 0x40013ef0 Remote: 0
                Label Stack: 16012  dpa-rev:35993054.  <<< LU Label>>>>
        Path:1  fec index: 0x2001f8b5(129205) DSP fec index: 0x200001f9(505),
                MPLS encap key: 0xf1b0000040013ef2 MPLS encap id: 0x40013ef2 Remote: 0
                Label Stack: 16012  dpa-rev:35993055  <<< LU Label>>>>

 REC-SHLDI HAL PD context :
ecd_marked:10, collapse_bwalk_required:0, load_shared_lb:0

    RSHLDI:
        PI:0x308dd32c38 PD:0x308dd32d08 rev:4927729 dpa-rev:35005343 flag:0x3
        FEC key: 0x1a1c740000d71 fec index: 0x20000813(2067) num paths: 2
        p-rev:4926086
        Path:0 fec index: 0x2001eefd(126717) DSP fec index: 0x200001f8(504),
        Path:1 fec index: 0x2001eefe(126718) DSP fec index: 0x200001f9(505),
LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:1, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
```

```
    PI:0x308dde33b8 PD:0x308dde3458 rev:4924403 type: MPLS (2)
    LEAF location: LEM
    FEC key: 0

    LWLDI:
        PI:0x308b04ea58 PD:0x308b04ea98 rev:4924400 p-rev:4924389 4924389 4924389 4924389
 ldi type:IMP_EOS0_EOS1
        FEC key: 0x1a75340000d71 fec index: 0x0(0) num paths:4, bkup paths: 0
        IMP LDI: ECD_MARKED
        IMP pattern:3
        PI:0x308b04ea58 PD:0x308b04ea98 rev:4924400 p-rev:4924389 4924389 4924389 4924389

        FEC key: 0x1a74720000d71 fec index: 0x200001f8(504) num paths:4
        Path:0  fec index: 0x2001ee86(126598) DSP:0x21
                MPLS encap key: 0xf1b0000040015878 MPLS encap id: 0x40015878 Remote: 0
                Label Stack: 16005  dpa-rev:34999715
        Path:1  fec index: 0x2001ee87(126599) DSP:0x22
                MPLS encap key: 0xf1b000004001587a MPLS encap id: 0x4001587a Remote: 0
                Label Stack: 16005  dpa-rev:34999716
        Path:2  fec index: 0x2001ee88(126600) DSP:0xc000002
                MPLS encap key: 0xf1b0000040016980 MPLS encap id: 0x40016980 Remote: 0
                Label Stack: 16005  dpa-rev:34989935
        Path:3  fec index: 0x2001ee89(126601) DSP:0xc000003
                MPLS encap key: 0xf1b00000400157fc MPLS encap id: 0x400157fc Remote: 0
                Label Stack: 16005  dpa-rev:34989936

SHLDI:
            PI:0x30927740c8 PD:0x3092774198 rev:4924389 dpa-rev:34999705 flag:0x0
            FEC key: 0x1a75340000d71 fec index: 0x200001ff(511) num paths: 4 bkup paths: 0

            p-rev:4924311 4924329 8779 4920854
            Path:0 fec index: 0x2001ee8f(126607) DSP:0x21 Dest fec index: 0x0(0)
            Path:1 fec index: 0x2001ee90(126608) DSP:0x22 Dest fec index: 0x0(0)
            Path:2 fec index: 0x2001ee91(126609) DSP:0xc000002 Dest fec index: 0x0(0)
            Path:3 fec index: 0x2001ee92(126610) DSP:0xc000003 Dest fec index: 0x0(0)
TX-NHINFO:
                PI: 0x308dc51298 PD: 0x308dc51318 rev:4924311 dpa-rev:34994174 Encap hdl:
0x3091632e98
                Encap id: 0x40010003 Remote: 0 L3 int: 1670 flags: 0x3
                npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:1f

            TX-NHINFO:
                PI: 0x308dc51c20 PD: 0x308dc51ca0 rev:4924329 dpa-rev:34994264 Encap hdl:
0x30916332c8
                Encap id: 0x40010001 Remote: 0 L3 int: 1679 flags: 0x3
                npu_mask: 0x1 DMAC: d4:6d:50:7c:f9:4d

            TX-NHINFO:
                PI: 0x308dc51ff0 PD: 0x308dc52070 rev:8779 dpa-rev:61964 Encap hdl:
0x308e9f4980
                Encap id: 0x40010007 Remote: 0 L3 int: 1728 flags: 0x807
                npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:22

            TX-NHINFO:
                PI: 0x308dc51480 PD: 0x308dc51500 rev:4920854 dpa-rev:34989846 Encap hdl:
0x308e9f4db0
                Encap id: 0x40010005 Remote: 0 L3 int: 1727 flags: 0x807
                npu_mask: 0x1 DMAC: 40:55:39:11:37:39

 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:1, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
```

```
    PI:0x308dde35b8 PD:0x308dde3658 rev:4926089 type: MPLS (2)
    LEAF location: LEM
    FEC key: 0

    LWLDI:
        PI:0x308b04eb48 PD:0x308b04eb88 rev:4926086 p-rev:4924389 4924389 4924389 4924389
 ldi type:IMP_EOS0_EOS1
        FEC key: 0x1a75340000d71 fec index: 0x0(0) num paths:4, bkup paths: 0
        IMP LDI: ECD_MARKED
        IMP pattern:3
        PI:0x308b04eb48 PD:0x308b04eb88 rev:4926086 p-rev:4924389 4924389 4924389 4924389

        FEC key: 0x1a74820000d71 fec index: 0x200001f9(505) num paths:4
        Path:0  fec index: 0x2001ee81(126593) DSP:0x21
                MPLS encap key: 0xf1b000004001587c MPLS encap id: 0x4001587c Remote: 0
                Label Stack: 16006  dpa-rev:35002526
        Path:1  fec index: 0x2001ee82(126594) DSP:0x22
                MPLS encap key: 0xf1b000004001588a MPLS encap id: 0x4001588a Remote: 0
                Label Stack: 16006  dpa-rev:35002527
        Path:2  fec index: 0x2001ee83(126595) DSP:0xc000002
                MPLS encap key: 0xf1b0000040016964 MPLS encap id: 0x40016964 Remote: 0
                Label Stack: 16006  dpa-rev:34991843
        Path:3  fec index: 0x2001ee84(126596) DSP:0xc000003
                MPLS encap key: 0xf1b00000400157fe MPLS encap id: 0x400157fe Remote: 0
                Label Stack: 16006  dpa-rev:34991844
SHLDI:
            PI:0x30927740c8 PD:0x3092774198 rev:4924389 dpa-rev:34999705 flag:0x0
            FEC key: 0x1a75340000d71 fec index: 0x200001ff(511) num paths: 4 bkup paths: 0

            p-rev:4924311 4924329 8779 4920854
            Path:0 fec index: 0x2001ee8f(126607) DSP:0x21 Dest fec index: 0x0(0)
            Path:1 fec index: 0x2001ee90(126608) DSP:0x22 Dest fec index: 0x0(0)
            Path:2 fec index: 0x2001ee91(126609) DSP:0xc000002 Dest fec index: 0x0(0)
            Path:3 fec index: 0x2001ee92(126610) DSP:0xc000003 Dest fec index: 0x0(0)

            TX-NHINFO:
                PI: 0x308dc51298 PD: 0x308dc51318 rev:4924311 dpa-rev:34994174 Encap hdl:
0x3091632e98
                Encap id: 0x40010003 Remote: 0 L3 int: 1670 flags: 0x3
                npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:1f

            TX-NHINFO:
                PI: 0x308dc51c20 PD: 0x308dc51ca0 rev:4924329 dpa-rev:34994264 Encap hdl:
0x30916332c8
                Encap id: 0x40010001 Remote: 0 L3 int: 1679 flags: 0x3
                npu_mask: 0x1 DMAC: d4:6d:50:7c:f9:4d

            TX-NHINFO:
                PI: 0x308dc51ff0 PD: 0x308dc52070 rev:8779 dpa-rev:61964 Encap hdl:
0x308e9f4980
                Encap id: 0x40010007 Remote: 0 L3 int: 1728 flags: 0x807
                npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:22

            TX-NHINFO:
                PI: 0x308dc51480 PD: 0x308dc51500 rev:4920854 dpa-rev:34989846 Encap hdl:
0x308e9f4db0
                Encap id: 0x40010005 Remote: 0 L3 int: 1727 flags: 0x807
                npu_mask: 0x1 DMAC: 40:55:39:11:37:39
```

Support for DHCPv4 and DHCPv6 Client over BVI

The Support for DHCPv4 and DHCPv6 Client over the BVI feature allows you to configure DHCPv4 and DHCPv6 client on the Bridged Virtual Interface (BVI). You can configure a BVI, and request DHCP IPv4 or IPv6 address on the BVI. This allows your customer's device to have initial connectivity to your network without user intervention in the field. After the device is connected to your network, the customer devices can push a node-specific configuration with static IP addresses on a different BVI for customer deployment.

Configure DHCPv4 and DHCPv6 Client over BVI

Perform the following tasks to configure DHCPv4 and DHCPv6 client over BVI:

- Configure AC interface
- Configure L2VPN
- Configure BVI

Configuration Example

```
/* Configure AC interface */
Router# configure
Router(config)# interface tenGigE 0/5/0/1/1
Router(config-if)# bundle id 1 mode on
Router(config-if)# exit
Router(config)# interface Bundle-Ether1
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# interface bundle-ether 1.100 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 100
Router(config-l2vpn-subif)# rewrite ingress tag pop 1 symmetric
Router(config-l2vpn-subif)# commit

/* Configure L2VPN */
Router # configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group BVI
Router(config-l2vpn-bg)# bridge-domain bvi
Router(config-l2vpn-bg-bd)# interface Bundle-Ether1.100
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)# routed interface BVI1
Router(config-l2vpn-bg-bd-bvi)# commit

/* Configure BVI */
Router# configure
Router(config)# interface BVI1
Router(config-if)# ipv4 address dhcp
Router(config-if)# ipv6 address dhcp
Router(config-if)# commit
```

Running Configuration

This section shows the DHCPv4 and DHCPv6 client over BVI running configuration.

```
interface TenGigE0/5/0/1/1
```

```
bundle id 1 mode on
!
interface Bundle-Ether1
!
interface Bundle-Ether1.100 l2transport
encapsulation dot1q 100
rewrite ingress tag pop 1 symmetric
!
l2vpn
bridge group BVI
  bridge-domain bvi
   interface Bundle-Ether1.100
   !
   routed interface BVI1
   !
  !
!
interface BVI1
ipv4 address dhcp
ipv6 address dhcp
!
```

Verification

The show output given in the following section display the details of DHCPv4 and DHCPv6 client over BVI configuration.

```
Router# show l2vpn bridge-domain
Legend: pp = Partially Programmed.
Bridge group: BVI, bridge-domain: bvi, id: 0, state: up, ShgId: 0, MSTi: 0
  Aging: 300 s, MAC limit: 64000, Action: none, Notification: syslog
  Filter MAC addresses: 0
  ACs: 2 (2 up), VFIs: 0, PWs: 0 (0 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of ACs:
    BV1, state: up, BVI MAC addresses: 1
    BE1.100, state: up, Static MAC addresses: 0
  List of Access PWs:
  List of VFIs:
  List of Access VFIs:

Router# show dhcp ipv4 client

    Interface name         IP Address      Binding State         Lease Time Rem
----------------------    ------------    ---------------    ----------------------
BVI1                      172.16.0.2      BOUND             3598 secs (00:59:58)
-------------------------------------------------------------------------------------
Router# show dhcp ipv6 client

    Interface name         IPv6 Address         State              Lease Time Rem
----------------------    --------------    ---------------     --------------------
BVI1                      2000::1           BOUND               2591982
-------------------------------------------------------------------------------------

Router# show dhcp ipv4 client bvi1 detail

-----------------------------------------------------
Client Interface name         : BVI1
Client Interface handle       : 0x8804054
Client ChAddr                 : 008a.9628.ac8a
Client ID                     : BVI1.00:8a:96:28:ac:8a
Client State                  : BOUND
Client IPv4 Address (Dhcp)    : 172.16.0.2
Client IPv4 Address Mask      : 255.240.0.0
```

```
Client Lease Time Allocated   : 3600 secs (01:00:00)
Client Lease Time Remaining   : 3571 secs (00:59:31)
Client Selected Server Address: 172.16.0.1
Client Next Hop Address       : 0.0.0.0
-------------------------------------------------

Router# show dhcp ipv4 client BVI1 statistics

Client Interface name           : BVI1
-------------------------------------------------
     CLIENT COUNTER(s)          |        VALUE
-------------------------------------------------
Num discovers sent              :         44
Num requests sent               :          1
Num offers received             :          1
Num acks received               :          1
-------------------------------------------------

Router# show dhcp ipv6 client

    Interface name         IPv6 Address         State              Lease Time Rem
----------------------    --------------    ---------------     --------------------
BVI1                       2000::1          BOUND               2591685
-------------------------------------------------------------------------------------

Router# show dhcp ipv6 client statistics-all

Interface name          : BVI1
Interface handle        : 0x8804054
VRF                     : 0x60000000

     TYPE         |    TRANSMIT    |      RECEIVE   |     DROP       |
-------------------------------------------------------------------------
SOLICIT           |            17  |             0  |             0  |
ADVERTISE         |             0  |             1  |             0  |
REQUEST           |             1  |             0  |             0  |
REPLY             |             0  |             2  |             0  |
CONFIRM           |             0  |             0  |             0  |
RENEW             |             1  |             0  |             0  |
REBIND            |             0  |             0  |             0  |
RELEASE           |             0  |             0  |             0  |
RECONFIG          |             0  |             0  |             0  |
INFORM            |             0  |             0  |             0  |


     TIMER         |    STARTED     |    STOPPED     |     EXPIRED    |
-----------------------------------------------------------------------
     INIT          |             1  |             0  |             1  |
     VBIND         |             0  |             0  |             0  |
     RENEW         |             2  |             1  |             0  |
     REBIND        |             2  |             1  |             0  |
     RETRANS       |            19  |             3  |            16  |
     VALID         |             2  |             1  |             0  |
```

Configure DHCPv6 Client Options

You can configure different DHCPv6 client options to differentiate between clients as required. Configure different DHCPv6 client options to differentiate how a DHCPv6 client communicates with a DHCPv6 server. The different DHCPv6 client options that you can configure are:

- **DUID:** If the DUID DHCPv6 client option is configured on an interface, DHCPv6 client communicates with the DHCPv6 server through the link layer address.

- **Rapid Commit:** If the Rapid Commit DHCPv6 client option is configured on an interface, DHCPv6 client can obtain configuration parameters from the DHCPv6 server through a rapid two-step exchange (solicit and reply) instead of the default four-step exchange (solicit, advertise, request, and reply).
- **DHCP Options:** The various other DHCPv6 options that can be configured on a DHCPv6 client are:
  - **Option 15:** Option 15 is also known as the User Class option and it is used by a DHCPv6 client to identify the type or category of users or applications it represents.
  - **Option 16:** Option 16 is also known as the Vendor ID option and it is used by a DHCPv6 a client to identify the vendor that manufactured the hardware on which the client is running.
  - **Option 23:** Option 23 is also known as the Domain name Server (DNS) option provides a list of one or more IPv6 addresses of DNS recursive name servers to which a client's DNS resolver can send DNS queries.
  - **Option 24:** Option 24 is also known as the Domain List option and it specifies the domain search list that the client uses to resolve hostnames with the DNS.
- **DHCP Timers:** This option is used to set different timer value for DHCP client configurations. The various DHCP timer options are:
  - **Release-timeout:** It is used to set retransmission timeout value for the initial release message.
  - **Req-max-rt:** It is used to set the maximum retransmission timeout value for the request message.
  - **Req-timeout:** It is used to set the initial request timeout value of the request message.
  - **Sol-max-delay:** It is used to set the maximum delay time of the first solicit message.
  - **Sol-max-rt:** It is used to set the maximum solicit retransmission time.
  - **Sol-time-out:** It is used to set the intial timeout value of the solicit message.

Configuration Example

Perform this task to configure DHCPv6 client options on a BVI interface.

```
Router# configure
Router(config)# interface BVI 10
Router(config-if)# ipv6 address dhcp-client-options
Router(config-dhcpv6-client)# duid linked-layer-address
Router(config-dhcpv6-client)# rapid-commit
Router(config-dhcpv6-client)# timers release-timeout 3
Router(config-dhcpv6-client)# timers sol-max-delay 1
Router(config-dhcpv6-client)# timers sol-time-out 1
Router(config-dhcpv6-client)# timers sol-max-rt 120
Router(config-dhcpv6-client)# timers req-max-rt 30
Router(config-dhcpv6-client)# timers req-timeout 1
Router(config-dhcpv6-client)# commit
```

Verification

To verify the DHCPv6 client options, use the **show dhcp ipv6 client BVI10 detail** command.

```
Router# show dhcp ipv6 client BVI10 detail
Wed Jun  10 16:19:21.272 IST

-------------------------------------------------------
```

```
Client Interface name : MgmtEth0/0/CPU0/1
Client Interface handle : 0x4040
Client MACAddr : 02f0.2b39.44be
Client State : BOUND
Client Link Local Address : fe80::f0:2bff:fe39:44be
Client IPv6 Address (Dhcp) : 600:1::12
Lease Remaining (in secs) : 74
DUID : 0003000102f02b3944be

Client Configuration
Timers
SOL_MAX_DELAY : 1 secs (00:00:01)
SOL_TIMEOUT : 1 secs (00:00:01)
SOL_MAX_RT : 120 secs (00:02:00)
REQ_TIMEOUT : 1 secs (00:00:01)
REQ_MAX_RT : 30 secs (00:00:30)
REL_TIMEOUT : 3 secs (00:00:01)

Options
RAPID-COMMIT : True
USER-CLASS : ciscoupnnp
VENDOR-CLASS : vendor
DNS-SERVERS : True
DOMAIN-LIST : True

DUID Type : DUID_LL

Server Information
Server Address : fe80::d2:a1ff:feb2:3b9f
Preference : 0
DUID : 000300010206826e2e00
Status : SUCCESS
IA-NA
Status : SUCCESS
IAID : 0x40400001
T1 : 60 secs (00:01:00)
T2 : 96 secs (00:01:36)
IA-ADDR
IA NA Address : 600:1::12
Preferred Time : 120 secs (00:02:00)
Valid Time : 120 secs (00:02:00)
Flags : 0x0
```

Related Topics

- Support for DHCPv4 and DHCPv6 Client over BVI, on page 106

Associated Commands

- show l2vpn bridge-domain
- show dhcp ipv4 client
- show dhcp ipv6 client
- show dhcp ipv4 client bvi

Layer 2 fast reroute

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Layer 2 Fast Reroute	Release 7.3.1	In the event of a link failure, this feature enables the router to switch traffic quickly to a precomputed loop-free alternative (LFA) path by allocating a label to the incoming traffic. This minimizes the traffic loss ensuring fast convergence. This feature introduces the **convergence reroute** command.

The L2 FRR feature safeguards against traffic loss when a PE-CE link fails before the remote PE receives the mass withdraw message indicating the link is down. On the PE router with the failed link to the CE, traffic is redirected through a backup path to its EVPN Multi-home (MH) peer router, which then forwards the traffic to the CE.

In the control plane, the MAC address remains associated with the local bridge port ESI, but in the data-path L2FRR activates the backup path for the MAC address which has been pre-populated on the AC segment.

As a consequence, **show** commands keep displaying the MAC address - bridge port association even after the AC is down.

Through this permanent association of hosts (or MAC addresses) to an AC or bridge port, the L2 MAC-IP routes are retained on PE1 even on failure. In addition to displaying the retained MAC address - bridge port association, the **show** commands on PE1 continues to display the retained ARP entries and L2 MAC-IP routes. The AC service state displays the **Down** state.

Managing MAC on layer 2 FRR-enabled attachment circuits (AC)

MAC addresses remain intact when a layer 2 FRR-enabled attachment circuit goes down.

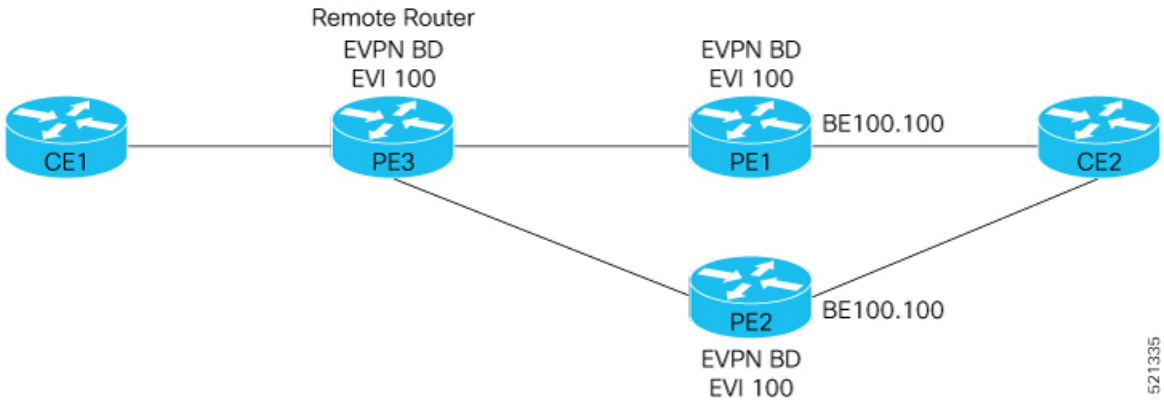
In ELAN configurations, retaining locally learned MAC addresses on layer 2 FRR-enabled circuits during downtime is crucial. This allows for quick identification and redirection of unicast traffic to the customer edge (CE) through MAC lookup.

Benefits

This feature provides fast and predictable convergence:

- Fast and predictable convergence.
- Manual configuration for predictable failover behavior.
- No need to change the topology.

Figure 14: Layer 2 Fast Reroute



In this topology:

- CE2 is multi-homed to PE1 and PE2.
- PE1 and PE2 are in EVPN active-active or single-active mode. They are connected to a remote router PE3 over the MPLS core network.
- CE1 is connected to PE3.
- Both PE1 and PE2 are enabled with L2 FRR. An FRR label is added per EVI for the backup path.

Consider a traffic flow from CE1 to CE2 in a regular scenario:

- The traffic is sent from CE1 to PE3.
- PE3 distributes the traffic over PE1 and PE2.
- PE1 and PE2 send the traffic to CE2.

When the PE3, CE2 traffic over PE1 takes the PE2 path which is PE1-PE2 backup path the traffic is forwarded to CE2. This traffic redirect lasts until PE3 updates the routes and sends to PE2.

- When L2FRR is enabled on PE1 and PE2, both PE1 and PE2 have the FRR labels. The logical backup path is pre-programmed in the hardware. When PE1 detects a failure on the access side (CE2), instead of sending traffic to CE2, PE1 sends the traffic over the pre-programmed backup path to PE2.
- All incoming traffic to PE1 is redirected to PE2 using this FRR label.
- PE1 encapsulates all the traffic with the FRR label of PE2 and forwards the traffic to PE2.
- Upon receiving the FRR label, PE2 forwards the traffic to CE2 through the attachment circuit (AC), even if the AC is in a blocking state.

Associated Commands

- **convergence reroute**
- **show evpn ethernet-segment**
- **show evpn evi**
- **show evpn evi ead private**

Restrictions for layer 2 fast reroute

This feature is supported on:

- BGP MPLS-based EVPN E-LAN.
- PE devices in EVPN active-active or single-active mode.
- Unicast traffic.
- This feature is not supported on BUM traffic.

Configure layer 2 fast reroute for E-LAN service

Configure L2 FRR on a PE router in the E-LAN EVPN multi-homing network.

Procedure

Step 1 Associate the Ethernet segment with the bundle interface and enable L2 FRR using the **convergence reroute** command. Configure this command on both PE1 and PE2 with in the specified topology map to enable L2FRR.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether4.1
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)# evi 1
Router(config-l2vpn-bg-bd-evi)# exit
Router(config-l2vpn-bg-bd)# exit
Router(config-l2vpn-bg)# exit
Router(config-l2vpn)# bridge group bg2
Router(config-l2vpn-bg)# bridge-domain bd2
Router(config-l2vpn-bg-bd)# interface Bundle-Ether4.2
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 2
Router(config-l2vpn-bg-bd-evi)# exit
Router(config-l2vpn-bg-bd)# exit
Router(config-l2vpn-bg)# exit
Router(config-l2vpn)#
```

```
Router# configure
Router(config)# evpn
Router(config-evpn)# evi 1
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance-mac)# exit
Router(config-evpn-instance)# exit
Router(config-evpn)# evi 2
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance-mac)# exit
Router(config-evpn-instance)# exit
Router(config-evpn)# interface Bundle-Ether4
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 40.00.00.00.00.00.00.00.01
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# convergence
Router(config-evpn-ac-es-conv)# reroute
Router(config-evpn-ac-es)# exit
```

```
Router(config-evpn-ac# exit
Router(config-evpn)# exit
Router(config)# exit
```

Step 2 Verify the L2 FRR configuration.

```
Router# show evpn ethernet-segment carving detail
...Ethernet Segment Id      Interface                         Nexthops
----------------------- ---------------------------------- --------------------
0040.0000.0000.0000.0001 BE4                                4.5.6.7
                                                            5.6.7.8
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether4
     Interface MAC  : 00c9.c654.9a04
     IfHandle       : 0x7800008c
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0040.0000.0000.0000.0001
  ES Import RT      : 4000.0000.0000 (from ESI)
  Topology          :
     Operational    : MH, Single-active
     Configured     : Single-active (AApS)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence    : Reroute
  Peering Details   : 2 Nexthops
     4.5.6.7 [MOD:P:00:T]
     5.6.7.8 [MOD:P:00:T]
  Service Carving Synchronization:
     Mode           : NTP_SCT
     Peer Updates   :
                 4.5.6.7 [SCT: 2025-01-22 17:01:01.1737583]
                 5.6.7.8 [SCT: 2025-01-22 17:00:36.1737583]
  Service Carving Results:
     Forwarders     : 2
     Elected        : 1
            EVI E   :        2
     Not Elected    : 1
            EVI NE  :        1
  EVPN-VPWS Service Carving Results:
     Primary        : 0
     Backup         : 0
     Non-DF         : 0
  MAC Flush msg     : STP-TCN
  Peering timer     : 3 sec [not running]
  Recovery timer    : 30 sec [not running]
  Carving timer     : 0 sec [not running]
  Revert timer      : 0 sec [not running]
  HRW Reset timer   : 5 sec [not running]
  Local SHG label   : 24008
     IPv6_Filtering_ID : 1:16
  Remote SHG labels : 1
              24007 : nexthop 5.6.7.8
  Access signal mode: Bundle OOS
  ...

Router# show l2vpn forwarding interface BE4.1 private location 0/RP0/CPU0
Wed Jan 22 17:02:01.387 EST

Xconnect ID 0xc0000002
```

```
  Xconnect info:
    xcon_status=Up, xcon_bound=TRUE, switching_type=0, data_type=12
    xcon_name=

    Object: XCON
    Base info: version=0xaabbcc13, flags=0x3110, type=2, object_id=UNSPECIFIED, reserved=0


  AC info:
   xcon_id=0xc0000002, ifh=0x7800008c, subifh=0x78000096, ac_id=0, ac_type=21, status=Bound

    ac_mtu=1500, iw_mode=1, adj=150.0.0.120+Bundle-Ether4,
    r_aps_channel=FALSE, prot_exclusion=FALSE
    rg_id=0, ro_id=0x0000000000000000
    evpn internal label = None
    E-Tree = Root
    FXC local-switch AC xcid = 0x0 (Invalid)
    FXC local-switch PW xcid = 0xffffffff (Invalid)
    EVPN MP route flags = 0x0
    Statistics:
      packets: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent 0
      bytes: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent 0
      MAC move: 0
      packets dropped: PLU 0, tail 0
      bytes dropped: PLU 0, tail 0

    Object: AC
    Base info: version=0xaabbcc11, flags=0x0, type=3, object_id=0x10001000000002d8|v9,
reserved=0

    AC Backup info:
      VC label: 24004
      Local VC label: 0
      Backup Pseudowire XC ID: 0x0
      Statistics:
        packets: received 0, sent 0
        bytes: received 0, sent 0
        packets dropped: PLU 0, tail 0, out of order 0
        bytes dropped: PLU 0, tail 0, out of order 0

      Object: AC_BACKUP
      Base info: version=0xaabbcc39, flags=0x0, type=43, object_id=0x1000100000000300|v1,
reserved=0


      Nexthop info:
        nh_addr=5.6.7.8,
        ecd_plat_data_valid=TRUE, ecd_plat_data_len=104, plat_data_size=232
        child_count=0, child_evpn_ole_count=2, child_mac_count=0, child_pwhe_mp_count=0,
child_ac_backup_count=2,
        child_vni_count=0, child_ifl_count=0, child_sg_count=0

        Object: NHOP
       Base info: version=0xaabbcc14, flags=0x4010, type=7, object_id=0x10001000000002f4|v5,
 reserved=0


  Bridge port info:
   xcon_id=0xc0000002, bridge_id=0, shg_id=0, mac_limit=disabled, 262144, mac_limit_action=0

    aging_timeout=300, mvrp_seq_number=0, learn_key=0
    bp_segl_type=0x3, mtu=1500
    is_flooding_disabled=FALSE, is_mac_learning_disabled=FALSE,
is_mac_port_down_flush_disabled=FALSE,
```

```
   EVPN ESI ID: 0
   EVPN SHG Local Label: None
   EVPN SHG Remote Labels: 0
    MAC learning: enabled
    Software MAC learning: enabled
    MAC port down flush: enabled
    Flooding:
      Broadcast & Multicast: enabled
      Unknown unicast: enabled
    MAC aging time: 300 s, Type: inactivity
    MAC limit: none
    MAC Secure: disabled, Logging: disabled, Accept-Shutdown: enabled
    DHCPv4 snooping: profile not known on this node, disabled
    Dynamic ARP Inspection: disabled, Logging: disabled
    IP Source Guard: disabled, Logging: disabled
    IGMP snooping profile: profile not known on this node
    MLD snooping profile: profile not known on this node
    Router guard disabled
    vES:disabled
    Etree Leaf:disabled
    STP participating: disabled
    Storm control: disabled
    Main port: Bundle-Ether4, MSTI: 2

    Object: BRIDGE_PORT
    Base info: version=0xaabbcc1a, flags=0x0, type=12, object_id=0x10001000000002d9|v6,
reserved=0
```

EVPN Preferred Nexthop

The EVPN Preferred Nexthop feature allows you to choose a primary nexthop and backup nexthop among the remote PE devices in dual-homed mode. By default, in an all-active dual-homed topology, traffic is load balanced using ECMP across both remote PE devices.

Configure the **preferred-nexthop** command when you want to direct traffic to one specific remote PE, and you are unable to use single-active mode due to hardware, topology, or technological limitations. The router allocates an internal label and will not allocate or consume ECMP FEC. The internal label enables fast switchover to backup PE when the primary link fails.

When remote PEs are operating in EVPN all-active mode, configure the **preferred-nexthop** command per EVI to choose an active and backup path based on the nexthop IP address. You can set the highest IP address as primary, which results in the lower IP address as a backup or vice versa.This feature provides you greater control over traffic patterns, that is to achieve symmetric traffic flow, and to allow support when a topology cannot support an all-active remote PE. Preferred nexthop is supported for native EVPN, EVPN VPWS, and EVPN PWHE. This feature supports a topology that has only two remote nexthops.

Configure EVPN Preferred Nexthop

Perform the following task to configure EVPN preferred nexthop.

Configuration Example

This example shows the configuration of highest IP address as the preferred nexthop.

```
Router# configure
Router(config)# evpn
```

```
Router(config-evpn)# evi 100
Router(config-evpn-evi)# preferred-nexthop highest-ip
Router(config-evpn-evi)# commit
```

This example shows the configuration of lowest IP address as the preferred nexthop.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# evi 100
Router(config-evpn-evi)# preferred-nexthop lowest-ip
Router(config-evpn-evi)# commit
```

This example shows the configuration of preferred nexthop using the **modulo** keyword.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# evi 100
Router(config-evpn-evi)# preferred-nexthop modulo
Router(config-evpn-evi)# commit
```

Running Configuration

This section shows the EVPN preferred nexthop running configuration.

```
/* Configuration of highest IP address as the preferred nexthop */
evpn
 evi 100
  preferred-nexthop highest-ip
!

/* Configuration of lowest IP address as the preferred nexthop */
evpn
 evi 100
  preferred-nexthop lowest-ip
!

/* Configuration of preferred nexthop using the modulo keyword */
evpn
 evi 100
  preferred-nexthop modulo
```

Verification

The output shows that the Highest IP is selected as primary (P) and the lowest IP as backup (B). The path selection is programmed in CEF.

```
Router#show evpn evi vpn-id 100 detail
Mon Oct 26 14:00:51.459 EDT

VPN-ID     Encap      Bridge Domain                 Type
---------- ---------- ----------------------------- -------------------
100        MPLS       bd100                         EVPN
…
   Preferred Nexthop Mode: Highest IP

Router#show evpn internal-label vpn-id  100 detail
Mon Oct 26 14:01:46.665 EDT

VPN-ID     Encap  Ethernet Segment Id         EtherTag     Label
```

```
---------- ------ -------------------------- ----------   --------
100        MPLS   0100.0000.acce.5500.0100    0            28120
      Multi-paths resolved: TRUE (Remote all-active) (Preferred NH, Highest IP)
      Multi-paths Internal label: 28120
      EAD/ES      192.168.0.1                                 0
                            192.168.0.3                                 0
      EAD/EVI     192.168.0.1                                 28099
                       192.168.0.3                                 28099
      Summary pathlist:
     0xffffffff (P) 192.168.0.3                                  28099
      0xffffffff (B) 192.168.0.1                                  28099

Router#show cef mpls local-label 28120 eOS
Mon Oct 26 14:04:10.851 EDT
Label/EOS 28120/1, version 56, internal 0x1000001 0x30 (ptr 0x4d3ba2a8) [1], 0x0 (0x0),
0x208 (0x4e6502c0)
Updated Oct 26 14:00:31.225
…
   via 192.168.0.3/32, 6 dependencies, recursive [flags 0x0]
    path-idx 0 NHID 0x0 [0x4d3bb58c 0x0], Internal 0x4e7890f8
    recursion-via-/32
    next hop 192.168.0.3/32 via 28103/0/21
     local label 28120
     next hop 27.27.27.3/32 Gi0/2/0/7    labels imposed {ImplNull 28099}
   via 192.168.0.1/32, 6 dependencies, recursive, backup (Local-LFA) [flags 0x300]
    path-idx 1 NHID 0x0 [0x4d3bb454 0x0]
    recursion-via-/32
    next hop 192.168.0.1/32 via 28105/0/21
     local label 28120
     next hop 26.26.26.1/32 Gi0/2/0/6    labels imposed {ImplNull 28099}
```

EVPN Access-Driven DF Election

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
EVPN Access-Driven DF Election	Release 7.3.1	This feature enables the access network to control EVPN PE devices by defining the backup path much before the event of a link failure, thereby reducing the traffic loss. The following keywords are added to the **service-carving** command: • **preference-based** • **access-driven**

This feature includes a preference-based and access-driven DF election mechanism.

In a preference-based DF election mechanism, the weight decides which PE is the DF at any given time. You can use this method for topologies where interface failures are revertive. However, for topologies where an access-PE is directly connected to the core PE, use the access-driven DF election mechanism.

When access PEs are configured in a non-revertive mode, the access-driven DF election mechanism allows the access-PE to choose which PE is the DF.

Consider an interface in an access network that connects PE nodes running Multichassis Link Aggregation Control Protocol (mLACP) and the EVPN PE in the core. When this interface fails, there may be a traffic loss for a longer duration. The delay in convergence is because the backup PE is not chosen before failure occurs.

The EVPN Access-Driven DF Election feature allows the EVPN PE to preprogram a backup PE even before the failure of the interface. In the event of failure, the PE node will be aware of the next PE that will take over. Thereby reducing the convergence time. Use the *preference df weight* option for an Ethernet segment identifier (ESI) to set the backup path. By configuring the weight for a PE, you can control the DF election, thus define the backup path.

Restrictions

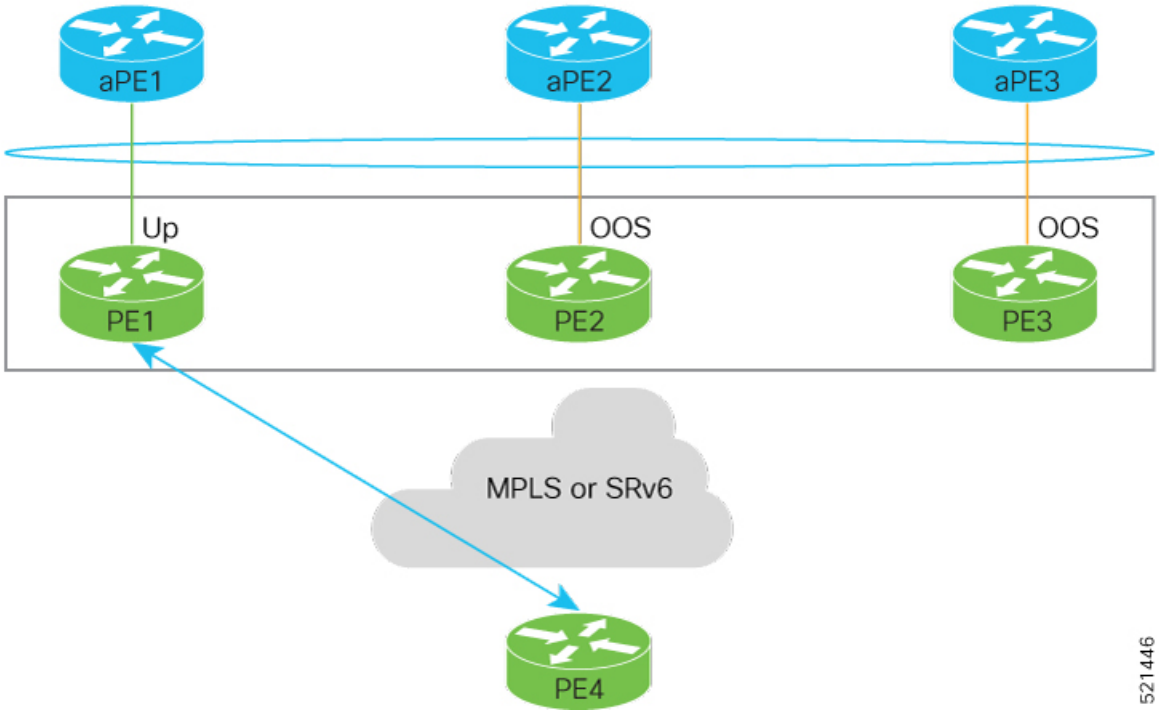
- The feature is supported only in an EVPN-VPWS scenario where EVPN PEs are in the port-active mode.
- The bundle attached to the ethernet segment must be configured with **lacp mode active**.

  LACP mode on is not supported.

Topology

Let's understand the feature on how the backup path is precomputed with the following topology.

Figure 15: EVPN Access-Driven DF Election

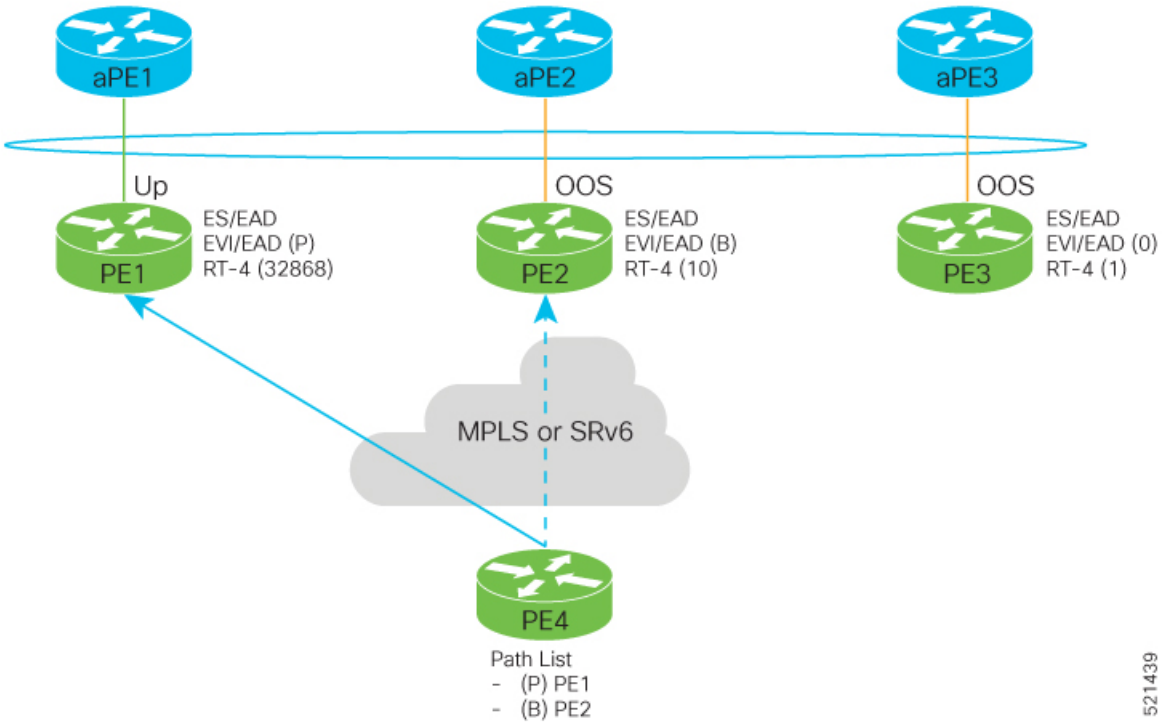


- PE1, PE2, and PE3 are PEs for the EVPN core network.
- aPE1, aPE2, and aPE3 are their access PE counterparts and configured in a multichassis link aggregation group (MCLAG) redundancy group. Only one link among the three is active at any given time. aPE1, aPE2, and aPE3 are in a non-revertive mode.

- PE1 is directly connected to aPE1, PE2 to aPE2, and PE3 to aPE3. EVPN VPWS is configured on the PE devices in the core.
- All PE devices are attached to the same bundle and shares the same ethernet segment identifier.
- PE1, PE2, and PE3 are configured with a weight of 100, 10, and 1 respectively.

Traffic Flow

In this example, consider a traffic flow from a host connected to PE4 to the host connected to the access PE.



- aPE1-PE1 interface state is up. The aPE2-PE2 and aPE3-PE3 remains in OOS state.
- The traffic is sent from PE4 to aPE1 through PE1 as the PE1 is configured with a highest weight of 100.
- The highest weight is modified by adding 32768 to the configured weight. For example, the weight of PE1 is 100, 32768 is added to this weight. Hence, 32868 is advertised to the peer PEs.
- The highest weight is advertised as P-bit, which is primary. The next highest weight is advertised as B-bit, which is secondary. The lowest weight as non-DF (NDF).
- When the EVPN PE devcies are of same weight, the traffic is sent based on the IP address. Lowest IP address takes the precedence.
- Only one PE indicates that the state of the bundle for the Ethernet Segment is up. For all other PEs, the Ethernet Segment is standby and the bundle is in OOS state.
- All PE devices are aware of the associated next hop and weights of their peers.

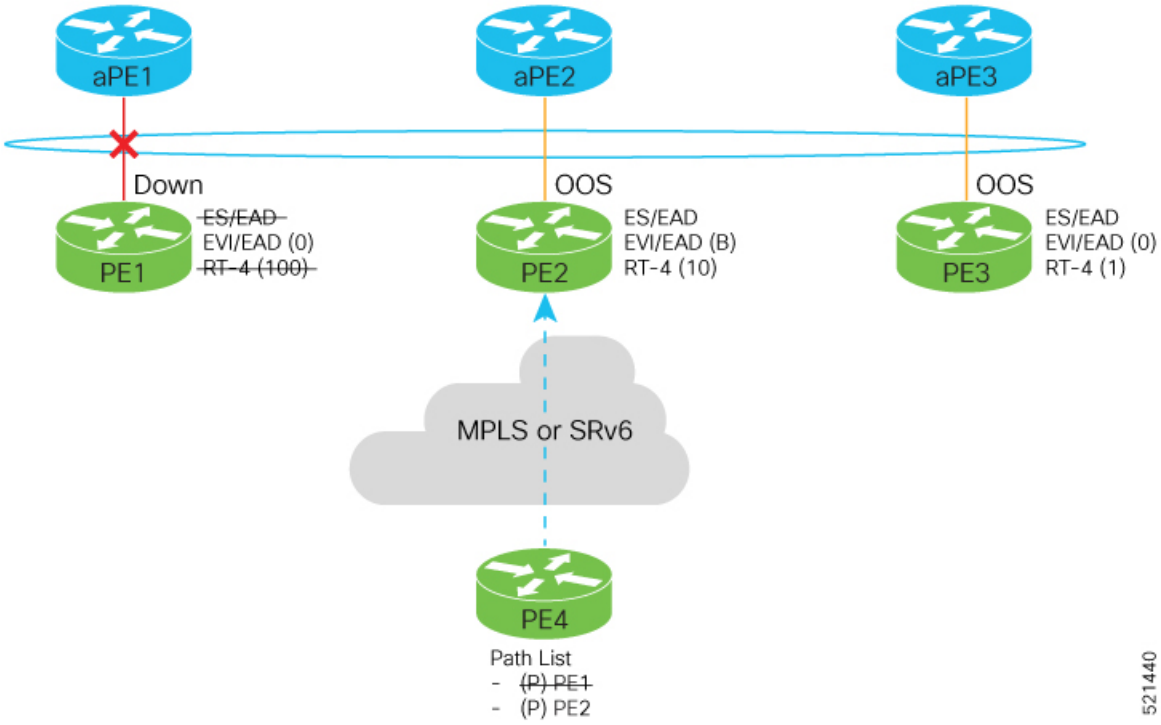
Failure and Recovery Scenarios

The weights configured on the EVPN PE devices cascade in the same order as the protection mechanism on the access side PEs:

- During the network failure, the redundancy ordering for the access PEs is aPE1, aPE2, aPE3.
- The weights of PE1 through PE3 are weight of PE1 > weight of PE2 > weight of PE3.
- If this ordering is not satisfied, the network will eventually converge, but it will not be as efficient as if the weights are ordered correctly.

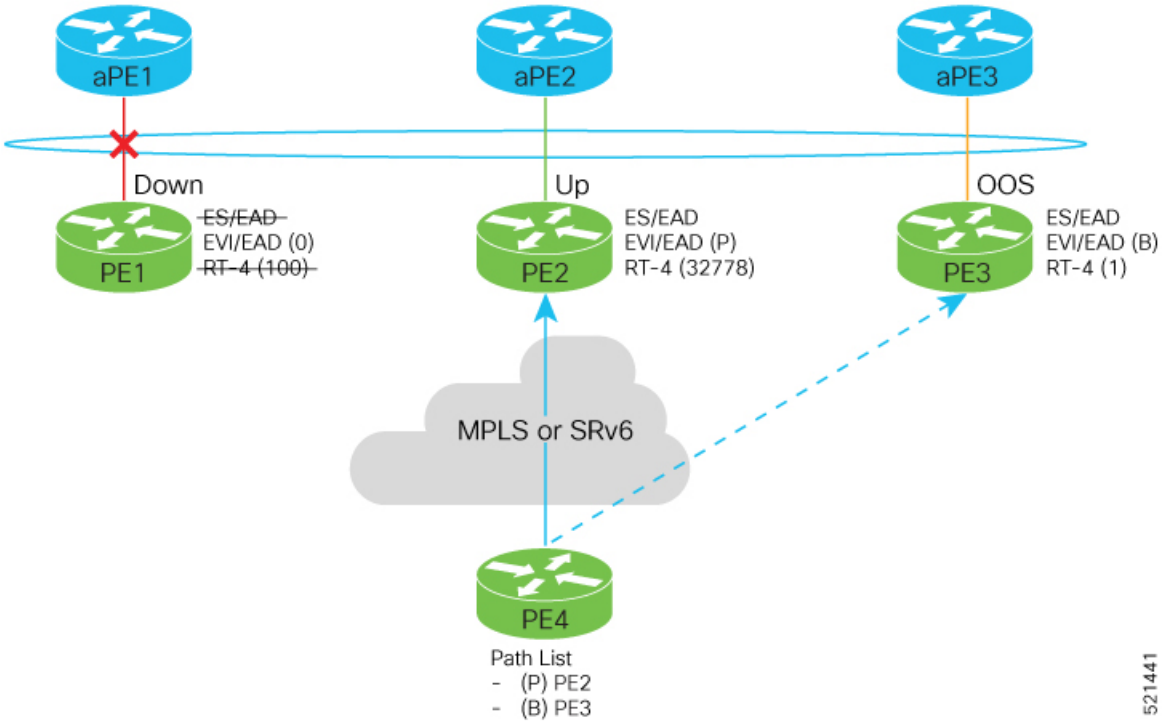
Scenario - 1

Consider a scenario where the aPE1-PE1 interface is down.



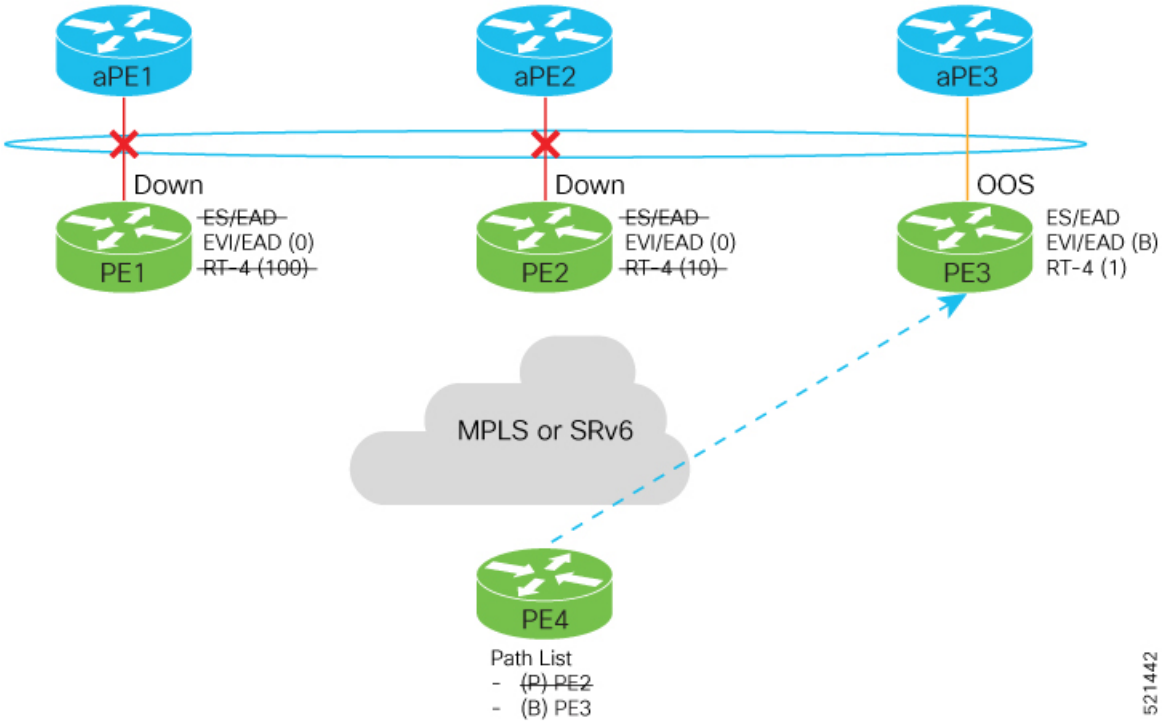
When aPE1-PE1 interface is down, the PE1 withdraws the EAD/ES route, and the traffic is sent through the backup path, which is PE2.

The aPE2-PE2 becomes the primary with a weight of 32778, and aPE3-PE3 becomes the backup. The aPE2-PE2 advertises P-bit to PE4. aPE3-PE3 advertises the B-bit to PE4.

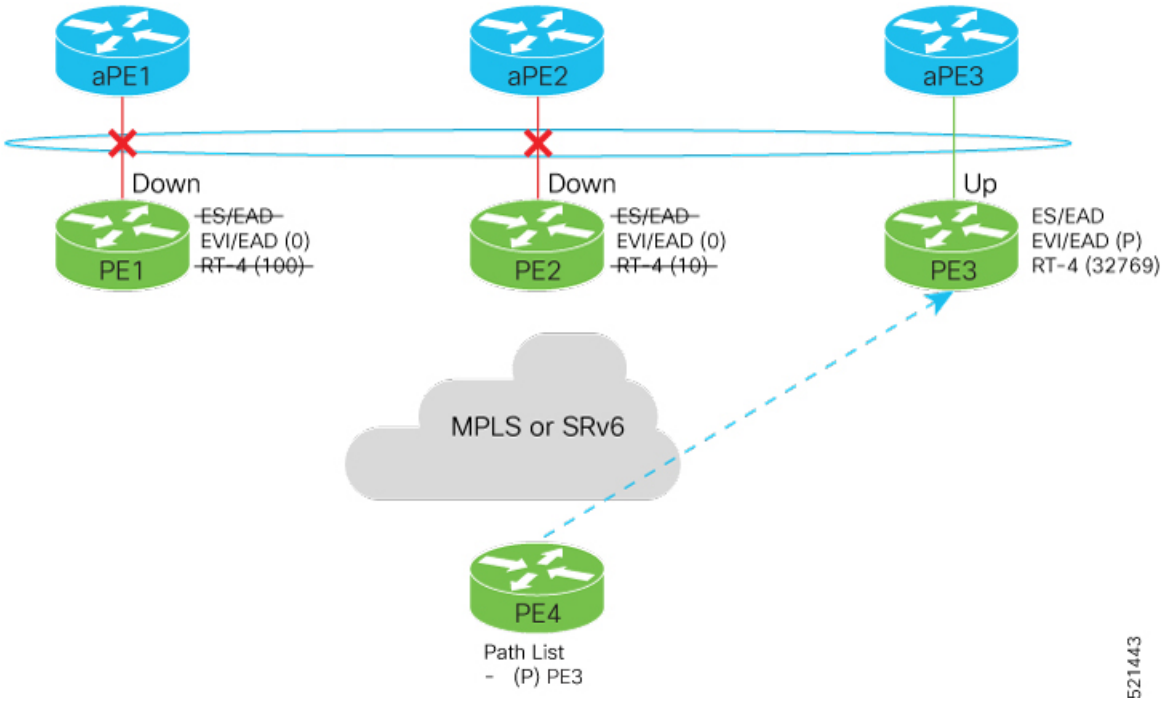


Scenario - 2

Consider a scenario where aPE2-PE2 interface is also down.

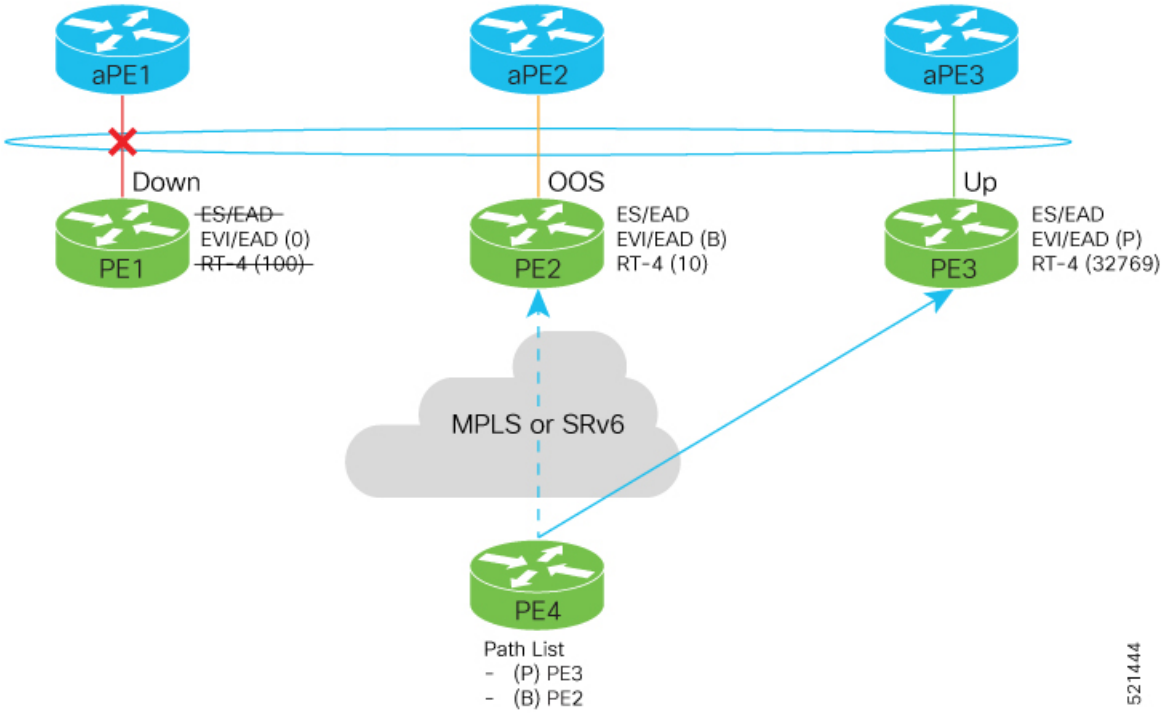


When the aPE2-PE2 interface is also down, the traffic is sent through aPE3-PE3 link. aPE3-PE3 becomes the primary path with a weight of 32769.



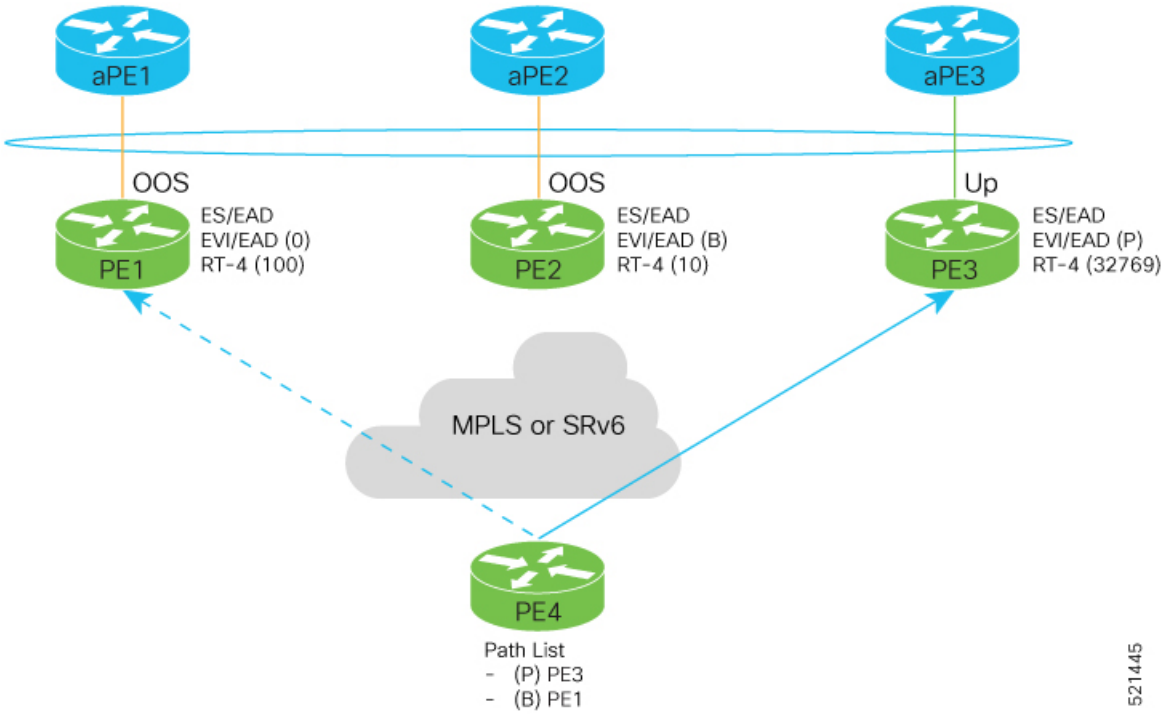
Scenario - 3

When the aPE2-PE2 interface comes up, the aPE3-PE3 link still remains the primary path. aPE2-PE2 interface becomes the backup path with a weight of 10.



Scenario - 4

When the aPE1-PE1 interface comes up, the aPE3-PE3 link remains the primary path with a weight of 32769. aPE1-PE1 interface becomes the backup path with a weight of 100. The aPE2-PE2 interface becomes NDF with a weight of 10.



Configure EVPN Access-Driven DF Election

Perform the following tasks to configure EVPN Access-Driven DF Election feature:

- Configure EVPN access-driven DF election on PE1, PE2, and PE3
- Configure LACP on aPE1, aPE2, and aPE3
- Configure EVPN-VPWS for PE1, PE2, and PE3

  See the *EVPN Virtual Private Wire Service (VPWS)* chapter on how to configure EVPN-VPWS.

Configuration Example

- All PE devices are configured with different weights. PE1, PE2, and PE3 are configured with a weight of 100, 10, and 1 respectively.
- The bundle attached to the ethernet segment is configured with **lacp mode active**.
- EVPN VPWS is configured on the PE devices.

```
/* Configure EVPN access-driven DF election on PE1, PE2, and PE3 */

/*  PE1 Configuration */
Router#configure
Router(config)#evpn
```

```
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment
Router(config-evpn-ac-es)#identifier type 0 01.11.00.00.00.00.00.00.01
Router(config-evpn-ac-es)#load-balancing-mode port-active
Router(config-evpn-ac-es)#service-carving preference-based
Router(config-evpn-ac-es-sc-pref)#weight 100
Router(config-evpn-ac-es-sc-pref)#access-driven
Router(config-evpn-ac-es-sc-pref)#commit

/* PE2 Configuration */
Router#configure
Router(config)#evpn
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment
Router(config-evpn-ac-es)#identifier type 0 01.11.00.00.00.00.00.00.01
Router(config-evpn-ac-es)#load-balancing-mode port-active
Router(config-evpn-ac-es)#service-carving preference-based
Router(config-evpn-ac-es-sc-pref)#weight 10
Router(config-evpn-ac-es-sc-pref)#access-driven
Router(config-evpn-ac-es-sc-pref)#commit

/* PE3 Configuration */
Router#configure
Router(config)#evpn
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment
Router(config-evpn-ac-es)#identifier type 0 01.11.00.00.00.00.00.00.01
Router(config-evpn-ac-es)#load-balancing-mode port-active
Router(config-evpn-ac-es)#service-carving preference-based
Router(config-evpn-ac-es-sc-pref)#weight 1
Router(config-evpn-ac-es-sc-pref)#access-driven
Router(config-evpn-ac-es-sc-pref)#commit
```

Configure LACP on aPE1, aPE2, and aPE3

```
/* aPE1 Configuration */
Router#configure
Router(config)#interface Bundle-Ether 1
Router(config-if)#lacp non-revertive
Router(config-if)#bundle maximum-active links 1 hot-standby
Router(config-if)#exit
Router(config-if)#interface GigabitEthernet0/0/0/40
Router(config-if)bundle id 10 mode active
Router(config-if)bundle port-priority 10000
Router(config-if)description Connection to PE1
Router(config-if)commit

/* aPE2 Configuration */
Router#configure
Router(config)#interface Bundle-Ether 1
Router(config-if)#lacp non-revertive
Router(config-if)#bundle maximum-active links 1 hot-standby
Router(config-if)#exit
Router(config-if)#interface GigabitEthernet0/0/0/39
Router(config-if)bundle id 10 mode active
Router(config-if)bundle port-priority 20000
Router(config-if)description Connection to PE2
Router(config-if)commit

/* aPE3 Configuration */
Router#configure
```

```
Router(config)#interface Bundle-Ether 1
Router(config-if)#lacp non-revertive
Router(config-if)#bundle maximum-active links 1 hot-standby
Router(config-if)#exit
Router(config-if)#interface GigabitEthernet0/0/0/38
Router(config-if)bundle id 10 mode active
Router(config-if)bundle port-priority 30000
Router(config-if)description Connection to PE3
Router(config-if)commit
```

Running Configuration

This section shows the running configuration of EVPN Access-Driven DF Election feature.

```
/* PE1 Configuration */
evpn
 interface Bundle-Ether 1
  ethernet-segment
   identifier type 0 01.11.00.00.00.00.00.00.01
    load-balancing-mode port-active
    service-carving preference-based
    weight 100
    access-driven
   !
 !

/* PE2 Configuration */
evpn
 interface Bundle-Ether 1
  ethernet-segment
   identifier type 0 01.11.00.00.00.00.00.00.01
    load-balancing-mode port-active
    service-carving preference-based
    weight 10
    access-driven
   !
 !

/* PE3 Configuration */
evpn
 interface Bundle-Ether 1
  ethernet-segment
   identifier type 0 01.11.00.00.00.00.00.00.01
    load-balancing-mode port-active
    service-carving preference-based
    weight 1
    access-driven
   !
 !

/* aPE1 Configuration */

interface Bundle-Ether 1
 lacp non-revertive
  bundle maximum-active links 1 hot-standby
interface GigabitEthernet0/0/0/40
 bundle id 10 mode active
 bundle port-priority 10000
 description Connection to PE1
!

/* aPE2 Configuration */
```

```
interface Bundle-Ether 1
 lacp non-revertive
  bundle maximum-active links 1 hot-standby
interface GigabitEthernet0/0/0/39
 bundle id 10 mode active
 bundle port-priority 20000
 description Connection to PE2
!

/* aPE3 Configuration */

interface Bundle-Ether 1
 lacp non-revertive
  bundle maximum-active links 1 hot-standby
interface GigabitEthernet0/0/0/40
 bundle id 10 mode active
 bundle port-priority 30000
 description Connection to PE3
!
```

Verification

Verify that you have configured the EVPN Access-Driven DF Election feature successfully.

```
Router#show evpn ethernet-segment detail
Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0001.0001.0001.1b01.001b BE1                               192.168.0.1
                                                           192.168.0.3
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether1
     Interface MAC  : 02ef.af8d.8008
     IfHandle       : 0x00004190
     State          : Up
     Redundancy     : Active
  ESI type          : 0
     Value          : 01.0001.0001.1b01.001b
  ES Import RT      : 0100.0100.011b (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH
     Configured     : Port-Active
  Service Carving   : Preferential
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 2 Nexthops
     192.168.0.1 [PREF:P:d6ce:T] >> Weight in hexadecimal
     192.168.0.3 [PREF:P:457]
  Service Carving Synchronization:
     Mode           : NONE
     Peer Updates   :
  Service Carving Results:
     Forwarders     : 24
     Elected        : 6
     Not Elected    : 0
  EVPN-VPWS Service Carving Results:
     Primary        : 18
     Backup         : 0
     Non-DF         : 0
  MAC Flushing mode : STP-TCN
```

```
Peering timer     : 3 sec [not running]
Recovery timer    : 30 sec [not running]
Carving timer     : 0 sec [not running]
Local SHG label   : 28384
Remote SHG labels : 0
Access signal mode: Bundle OOS (Default)
```

Associated Commands

- service-carving
- show evpn ethernet-segment