EVPN Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 7.11.x

First Published: 2023-12-01

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Preface

This guide describes the Cisco 8000 Series Router configurations. The preface for the EVPN Configuration Guide for Cisco 8000 Series Routers contains these sections:

Changes to This Document

This table lists the technical changes made to this document since it was first released.

Table 1: Changes to This Document

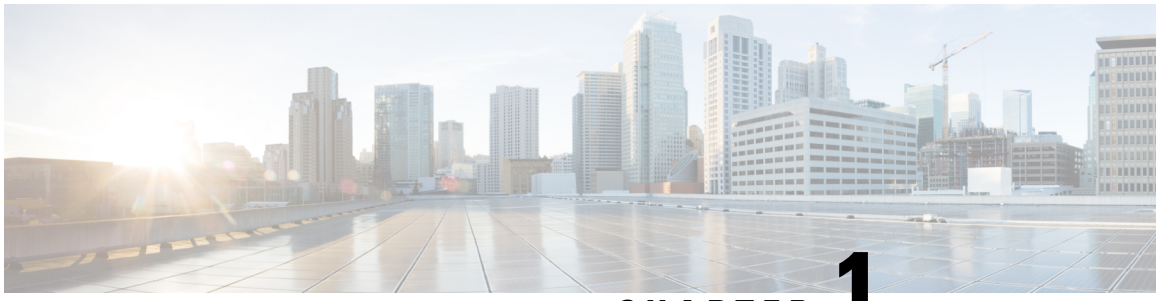
Date	Summary
December 2023	Initial release of this document

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly What's New in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the What's New in Cisco Product Documentation as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



CHAPTER 1

New and Changed EVPN Features

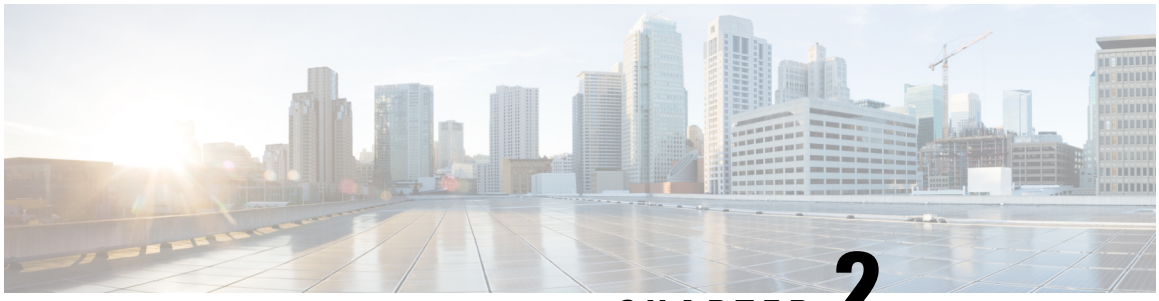
This table summarizes the new and changed feature information for the EVPN Configuration Guide for Cisco 8000 Series Routers, and tells you where they are documented.

New and Changed EVPN Features

Table 2: EVPN Features Added or Modified in IOS XR Release 7.11.x

Feature	Description	Changed in Release	Where Documented
EVPN E-LAN L2 Gateway Single-Homing	This feature was introduced.	Release 7.11.1	EVPN E-LAN L2 Gateway Single-Homing, on page 14

New and Changed EVPN Features



CHAPTER 2

YANG Data Models for EVPN Features

This chapter provides information about the YANG data models for EVPN features.

• Using YANG Data Models, on page 3

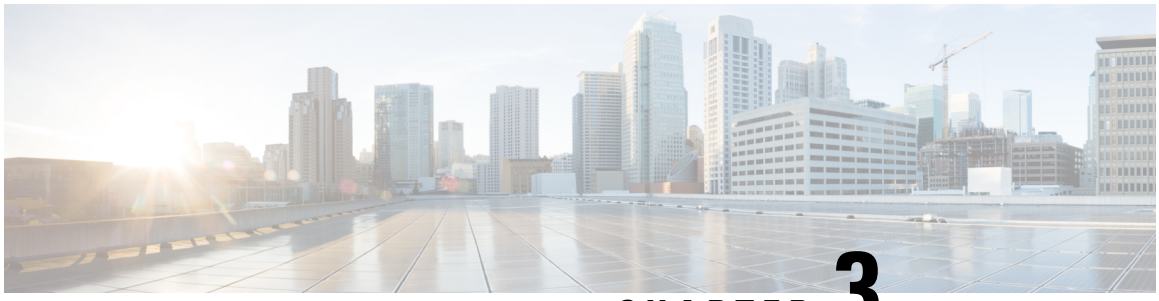
Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 3

Getting Started with EVPN

EVPN Overview

Today, our networks have different protocols serving different purposes, which makes daily operations more complex than they need to be. This hinders the ability to deliver end-to-end services with speed and agility. As you deploy multiple geographically disparate data centers, they're looking for scalable and simplified network solutions to extend virtualization and cluster domains between multiple data centers.

Ethernet VPN (EVPN) is the next-generation L2VPN technology, and it provides layer 2 and 3 VPN services in a scalable and simplified manner. The evolution of EVPN started due to the need for a scalable solution to bridge various layer 2 domains and overcome the limitations faced by VPLS, such as scalability, multihoming, and per-flow load balancing.

EVPN provides secure and private connectivity of multiple sites within an organization spread across different geographical locations. EVPN operates in contrast to the existing VPLS by enabling control-plane-based MAC learning. In EVPN, PEs participating in the EVPN instances learn customer MAC routes in the control plane using the MP-BGP protocol. EVPN brings various benefits addressing the VPLS shortcomings, including multi-homing support with per-flow load balancing. EVPN uses MAC addresses as routable addresses and distributes them to all participating PEs through the MP-BGP EVPN control plane.

To know more about EVPN, visit https://e-vpn.io.

EVPN supports E-LAN, E-LINE, E-TREE services, and provides data-plane and control-plane separation, and much more.

EVPN allows the use of different encapsulation mechanisms in the data plane while maintaining the same control plane. In addition, EVPN offers many advantages over existing technologies, including more efficient load-balancing of VPN traffic.

Benefits

- Per flow-based load balancing
- Scalability
- Reduced operational complexity
- Improved network efficiency by eliminating flooding and learning
- Provides fast reroute, resiliency, fast reconvergence during link failure
- Integrates L2 and L3 VPN services

EVPN Key Concepts

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

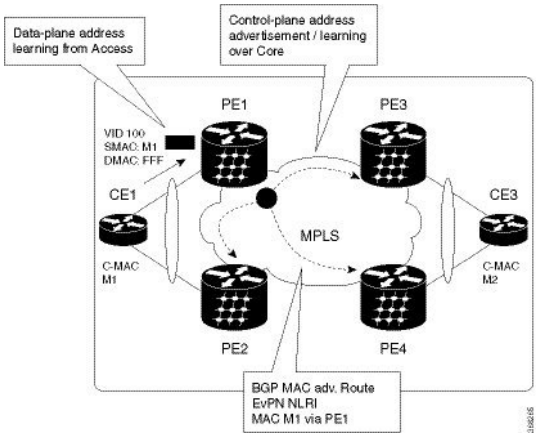
- Ethernet Segment (ES): An Ethernet segment is a set of Ethernet links that connects a multihomed device. If a multi-homed device or network is connected to two or more PEs through a set of Ethernet links, then that set of links is referred to as an Ethernet segment. The Ethernet segment route is also 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 0xFFFFFF 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 the 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 3: 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 Modes

The following EVPN modes are supported:

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

EVPN Timers

The following table shows various EVPN timers:

Table 4: EVPN Timers

Timer	Range	Default Value	Trigger	Applicability	Action	Sequence
startup-cost-in	30-86400	disabled	node recovered*	Single-Homed, All-Active, Single-Active	Postpone EVPN startup procedure and Hold AC link(s) down to prevent CE to PE forwarding. Startup-cost-in timer allows PE to set core protocols first.	1
recovery	20-3600s	30s	node recovered, interface recovered **	Single-Homed***, Single-Active	Postpone EVPN Startup procedure. Recovery timer allows PE to set access protocols (STP) before reachability towards EVPN core is advertised.	2
peering	0-3600s	3s	node recovered, interface recovered	All-Active, Single-Active	Starts after sending EVPN RT4 to postpone rest of EVPN startup procedure. Peering timer allows remote PE (multihoming AC with same ESI) to process RT4 before DF election will happen.	3

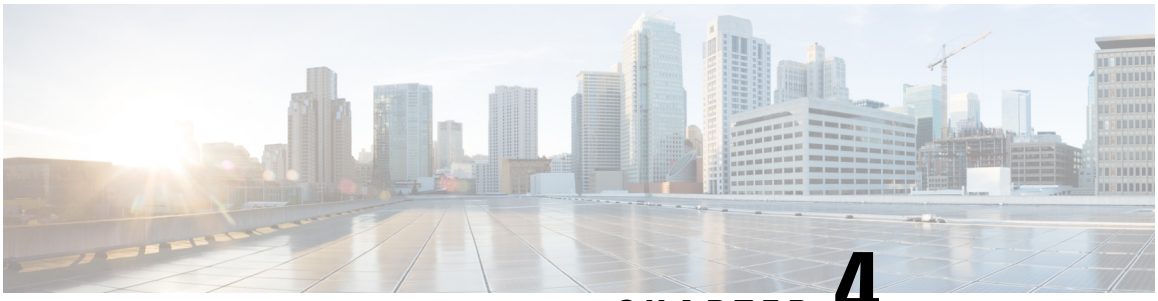
Note

- The timers are available in EVPN global configuration mode and in EVPN interface sub-configuration mode.
- Startup-cost-in is available in EVPN global configuration mode only.
- Timers are triggered in sequence (if applicable).
- Cost-out in EVPN global configuration mode brings down AC link(s) to prepare node for reload or software upgrade.

* indicates all required software components are loaded.

** indicates link status is up.

*** you can change the recovery timer on Single-Homed AC if you do not expect any STP protocol convergence on connected CE.



CHAPTER 4

EVPN E-LAN L2 Gateway Single-Homing

This chapter describes how to configure EVPN E-LAN L2 Gateway Single-Homing.

EVPN E-LAN L2 Gateway Single-Homing

Table 5: Feature History Table

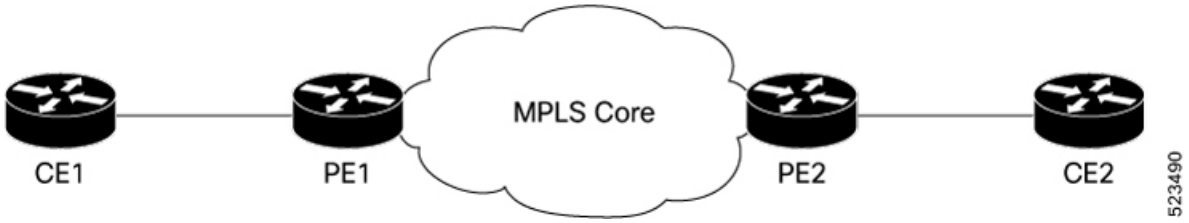
Feature Name	Release Information	Feature Description
EVPN E-LAN L2 Gateway Single-Homing	Release 7.11.1	We now offer a cost-effective and simplified solution for seamless communication between various customer sites connected to the same service provider network using Ethernet Virtual Private Network (EVPN) single-homing mode. EVPN LAN (E-LAN) is a service to bridge Ethernet data traffic among different sites across the MPLS network connecting a Layer 2 gateway device to a single access network. In the single-homing mode, a device is connected to one router in the MPLS core through physical ports or bundle ports, and in the event of a failure on those links, the traffic over the links is not protected by links to another router on the core. This feature is supported only on Q200-based line cards. The feature introduces the **evpn** commands.

Deploying EVPN single-homing can simplify network infrastructure management and scaling by requiring only one provider edge router for connectivity, resulting in significant benefits. Additionally, reducing the need for additional infrastructure through implementing EVPN single-homing can lead to substantial cost savings for the initial setup and ongoing maintenance.

The EVPN network provides a solution for linking a network or device to a single physical or bundle link. This approach does not come with built-in redundancy or automatic failover capabilities. Nevertheless, you can use various mechanisms to ensure high availability and minimize downtime through appropriate failover mechanisms like link or route redundancy. Evaluating your specific network requirements is essential when deploying EVPN single-homing. Although this option offers cost savings and simplicity, it may be better suited to smaller or medium-sized enterprises that require only a single connection to the EVPN network.

Topology

Using this topology, let’s understand how EVPN E-LAN Layer 2 gateway single-homing transports traffic from one customer site to another.



- CE1 is connected to PE1 using a single bundle or physical link. When you send Layer 2 traffic from CE1 to CE2, the traffic is encapsulated in Layer 2 frames.
- PE1 receives the Layer 2 frames on the ingress interface from CE1. PE1 checks the destination MAC address of the frame and determines the appropriate attachment circuit to forward the frame.
- PE1 then uses EVPN control plane protocols to distribute the MAC address information learned from CE1 to PE2.
- PE2 router that has the destination MAC address obtained from the EVPN control plane forwards the Layer 2 frames to the appropriate attachment circuit connected to CE2.

Configure EVPN E-LAN L2 Gateway Single -Homing

In this topology, configure EVPN E-LAN L2 Gateway single-homing on PE1. You must configure Ethernet VPN Identifier (EVI) under the bridge domain and enable PE1 to advertise MAC addresses to distribute the MAC address information learned from CE1 to PE2.

Perform the following tasks to configure EVPN E-LAN L2 gateway single-homing on PE1:

1. Set up BGP for L2VPN and EVPN
2. Configure bridge domain
3. Configure MAC advertisement

Configuration Example

```
/* Set up BGP for L2VPN and EVPN */
Router# configure
Router#(config)# router bgp 200
Router#(config-bgp)# bgp router-id 10.10.10.1
Router#(config-bgp)# address-family l2vpn evpn
Router#(config-bgp)# neighbor 10.10.10.10
Router#(config-bgp-nbr)# remote-as 200
Router#(config-bgp-nbr)# update-source Loopback 0
Router#(config-bgp-nbr)# address-family l2vpn evpn

/* Configure bridge domain */
Router(config)# l2vpn
Router (config-l2vpn)# bridge group BG1
Router (config-l2vpn-bg)# bridge-domain BD1
Router (config-l2vpn-bg-bd)# interface Bundle-Ether1.2001
Router (config-l2vpn-bg-bd-ac)# evi 2001

/* Configure MAC advertisement */
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether 1
```

```
Router(config-evpn-ac)# exit
Router(config-evpn)# evi 2001
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance-mac)# commit
```

Running Configuration

This section shows an EVPN E-LAN L2 gateway single-homing running configuration.

```
router bgp 200
 bgp router-id 10.10.10.1
 address-family l2vpn evpn
 neighbor 10.10.10.10
  remote-as 200 description MPLS-FACING-PEER
  update-source Loopback0
  address-family l2vpn evpn
!

l2vpn
 bridge group BG1
  bridge-domain BD1
   interface BundleEther1.2001
    evi 2001
!

evpn
 interface Bundle-Ether 1
!
evi 2001
 advertise-mac
!
```

Verification

Verify that EVPN E-LAN L2 gateway single-homing is configured.

In this example, the operational mode is SH or single-homing, which indicates that CE1 is connected to PE1 through a single link.

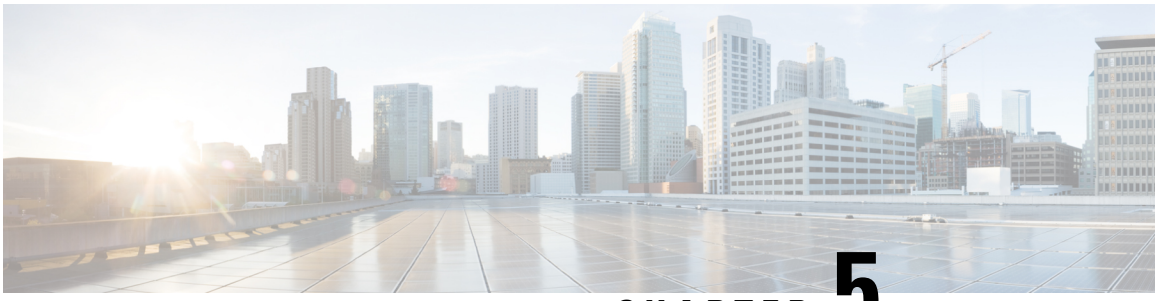
```
Router# show evpn ethernet-segment interface Bundle-Ether 1 detail

Ethernet Segment Id    Interface        Nexthops
---------------------  ----------       ----------
N/A                    Bundle-Ether 1   10.0.0.2
.....
Topology :
Operational : SH
```

EVPN E-LAN L2 Gateway Single-Homing Features

EVPN E-LAN L2 Gateway single-homing supports the following functionalities:

- BUM Ingress Replication for EVPN Single-Homing, on page 49
- Split-Horizon Groups for EVPN Single-Homing, on page 51
- VRF Leaking for EVPN Single-Homing, on page 53



CHAPTER 5

EVPN Virtual Private Wire Service (VPWS)

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet VPN Virtual Private Wire Service	Release 7.8.1	The Ethernet VPN Virtual Private Wire Service (EVPN-VPWS) is a BGP control plane solution for point-to-point services. It implements the signaling and encapsulation techniques for establishing an EVPN instance between a pair of PEs. It provides the service of forwarding L2 Ethernet traffic between network devices without inspecting the MAC header in the Ethernet frame. The use of EVPN for VPWS eliminates the need for signaling single-segment and multi-segment pseudowire (PW) for point-to-point Ethernet services.

The EVPN-VPWS technology works on IP and MPLS core; IP core to support BGP and MPLS core for switching packets between the endpoints.

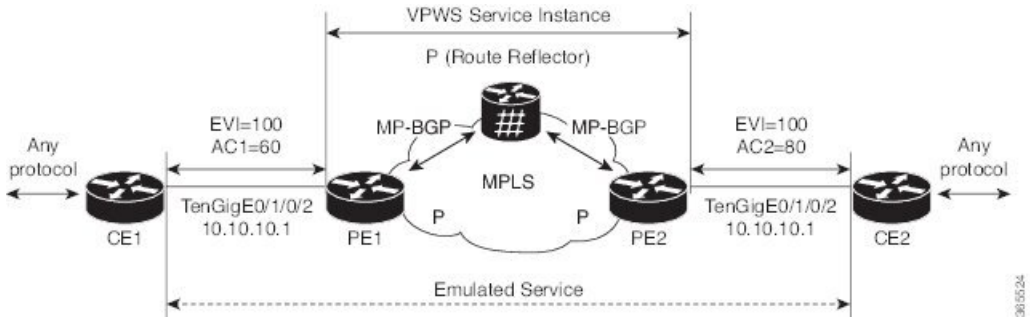
EVPN-VPWS Single Homed

The EVPN-VPWS single homed solution requires per EVI Ethernet Auto Discovery route. EVPN defines a new BGP Network Layer Reachability Information (NLRI) used to carry all EVPN routes. BGP Capabilities Advertisement used to ensure that two speakers support EVPN NLRI (AFI 25, SAFI 70) as per RFC 4760.

The architecture for EVPN-VPWS is that the PEs run Multi-Protocol BGP in a control-plane.

The following image describes the EVPN-VPWS configuration:

Figure 2: EVPN-VPWS Single Homed



- The VPWS service on PE1 requires the following three elements to be specified at the configuration time:
  - The VPN ID (EVI).
  - The local AC identifier (AC1) that identifies the local end of the emulated service.
  - The remote AC identifier (AC2) that identifies the remote end of the emulated service.

  PE1 allocates an MPLS label per local AC for reachability.
- The VPWS service on PE2 is set in the same manner as PE1. The three same elements are required and the service configuration must be symmetric.

  PE2 allocates an MPLS label per local AC for reachability.
- PE1 advertises a single EVPN per EVI Ethernet AD route for each local endpoint (AC) to remote PEs with the associated MPLS label.

  PE2 performs the same task.
- On reception of EVPN per EVI EAD route from PE2, PE1 adds the entry to its local EVPN data base. PE1 knows the path list to reach AC2, for example, next hop is PE2 IP address and MPLS label for AC2.

  PE2 performs the same task.

Restrictions for EVPN-VPWS

- EVPN-VPWS does not support Pseudowire Headend (PWHE) configuration.
- EVPN-VPWS is supported only on single-homing and is not supported on dual homing. This is applicable for both the local and remote sides of the network.

  EVPN validates if the route is for a single home next hop, otherwise it issues an error message. An Ethernet Segment Identifier (ESI) is an attribute that is used to enable EVPN multi-homing. EVPN relies on the ESI value being zero to determine if this is a single home or not. If the AC is a Bundle-Ether interface running LACP, then you need to manually configure the ESI value to zero to overwrite the auto-sense ESI, as EVPN-VPWS multi-homing is not supported.

  To disable EVPN dual homing, configure bundle-Ether AC with ESI value (**identifier type**) set to zero.

```
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether12
```

```
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.00.00.00.00.00.00.00.00
```

As an alternative, you can disable EVPN dual homing globally.

```
Router(config)# evpn
Router(config-evpn)# ethernet-segment type 1 auto-generation-disable
```

Configure EVPN-VPWS Single Homed

This section describes how to configure single-homed EVPN-VPWS feature.

```
/* Configure PE1 */
Router# configure
Router(config)# router bgp 100
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 10.10.10.1
Router(config-bgp-nbr))# address-family l2vpn evpn
Router(config-bgp-nbr-af)# commit
Router(config-bgp-nbr-af)# root
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn1
Router(config-l2vpn-xc-p2p)# interface TenGigE0/1/0/2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 100 target 12 source 10
Router(config-l2vpn-xc-p2p-pw)# exit
Router(config-l2vpn-xc-p2p)# commit

/* Configure PE2 */
Router# configure
Router(config)# router bgp 100
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 10.10.10.1
Router(config-bgp-nbr))# address-family l2vpn evpn
Router(config-bgp-nbr-af)# commit
Router(config-bgp-nbr-af)# root
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn1
Router(config-l2vpn-xc-p2p)# interface TenGigE0/1/0/2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 100 target 10 source 12
Router(config-l2vpn-xc-p2p-pw)# exit
Router(config-l2vpn-xc-p2p)# commit
```

If the source and target AC IDs are the same, use the following command to configure the neighbor EVPN:

neighbor evpn evi 100 service 10

Running Configuration

```
/* On PE1 */
configure
router bgp 100
 address-family l2vpn evpn
 neighbor 10.10.10.1
  address-family l2vpn evpn
```

```
!

configure
l2vpn
 xconnect group evpn-vpws
  p2p evpn1
   interface TenGigE0/1/0/2
   neighbor evpn evi 100 target 12 source 10
!


/* On PE2 */
configure
router bgp 100
 address-family l2vpn evpn
 neighbor 10.10.10.1
  address-family l2vpn evpn
!

configure
l2vpn
 xconnect group evpn-vpws
  p2p evpn1
   interface TenGigE0/1/0/2
   neighbor evpn evi 100 target 10 source 12
!
```

EVPN Seamless Integration with Legacy VPWS

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
EVPN Seamless Integration with Legacy VPWS	Release 7.8.1	When expanding an existing L2VPN network, users may want to deploy EVPN-VPWS to provide additional Layer 2 point-to-point Ethernet services, and at the same time some of their customer traffic may still need to be terminated on the existing L2VPN PEs on their network. Users can migrate the PE nodes from L2VPN VPWS to EVPN-VPWS, without disruption in traffic. The seamless migration offers users the option to use either VPWS or EVPN-VPWS services on PE nodes. This allows the coexistence of legacy VPWS and EVPN-VPWS dual-stack in the core for a given L2 Attachment Circuit (AC) over the same MPLS network. This feature introduces the **vpws-seamless-integration** command.

Although VPWS is a widely deployed Layer 2 VPN technology, some users prefer to migrate to EVPN service in their existing VPWS networks to leverage the benefits of EVPN services.

With EVPN-VPWS Seamless Integration feature, users can migrate the PE nodes from legacy VPWS service to EVPN-VPWS gradually and incrementally without any service disruption.

Users can migrate an Attachment Circuit (AC) connected to a legacy VPWS pseudowire (PW), which is using targeted-LDP signaling or BGP-AD signaling, to an EVPN-VPWS service.

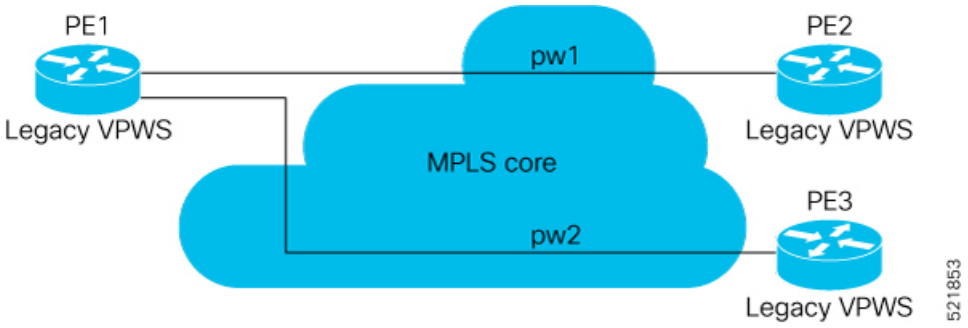
In an EVPN-VPWS network, VPN instances are grouped by EVPN Instance VPN ID (EVI) and identified by an ethernet tag or attachment circuit ID (AC-ID). EVI is also associated with route-targets and route-distinguisher.

During migration, an EVPN-VPWS PE router performs either VPWS or EVPN-VPWS L2 cross-connect for a given AC. When both EVPN-VPWS and BGP-AD PWs are configured for the same AC, the EVPN-VPWS PE during migration advertises the BGP VPWS Auto-Discovery (AD) route as well as the BGP EVPN Auto-Discovery (EVI/EAD) route and gives preference to EVPN-VPWS Pseudowire (PW) over the BGP-AD VPWS PW.

Let's understand how a legacy VPWS network can be migrated seamlessly to EVPN-VPWS with the following scenario:

Consider that a user plans to migrate VPWS node to an EVPN node one at a time. The user expects the migration to span over multiple years.

Figure 3: VPWS Nodes

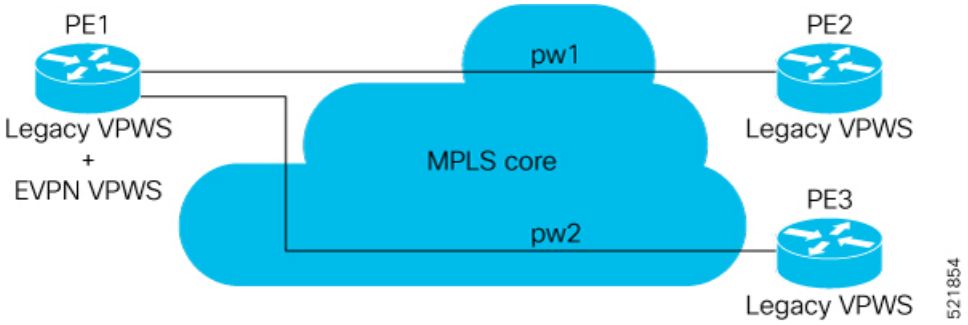


In this topology, PE1, PE2, PE3 are provider edge devices in the MPLS network and the legacy VPWS cross-connects are up and running between PE1, PE2, and PE3.

- PE1 and PE2 have a legacy PW established between them. (pw1)
- PE1 and PE3 have a legacy PW established between them. (pw2)

The user wants to replace PE1 with a new hardware. After replacing the equipment, the user enables EVPN-VPWS on PE1.

Figure 4: PE1 Enabled with EVPN-VPWS

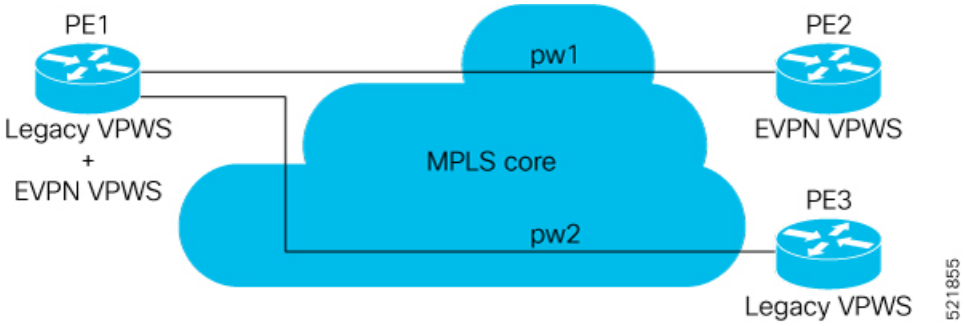


Let's understand what happens when only PE1 is migrated to EVPN-VPWS:

- When EVPN-VPWS is enabled, PE1 starts advertising EVPN EVI or Ethernet-AD route to other PE nodes.
- PE1 advertises BGP VPWS Auto-Discovery route and the BGP EVPN Ethernet-AD per EVI route for a given PW.
- As PE2 and PE3 aren't yet migrated, PE1 does not receive any EVI/EAD routes from these PE nodes. Therefore, legacy VPWS runs between PE1, PE2, and PE3.
- PE1 keeps forwarding traffic using legacy VPWS.

After one year, the user decides to upgrade PE2 and wants to migrate from VPWS to EVPN-VPWS.

Figure 5: PE2 enabled with EVPN-VPWS



- When the upgrade is completed, PE2 starts advertising EVI/EAD route to other PE nodes.
- Both PE1 and PE2 discover each other through EVPN routes.
- As a result, EVPN-VPWS service replaces legacy VPWS service between PE1 and PE2. This is called EVPN Seamless Integration with legacy VPWS.
- EVPN-VPWS service takes high-precedence over legacy VPWS network.
- PE1 and PE2 shuts down the legacy VPWS between them to prevent ongoing duplicate packets from remote CE.

PE3 device is not yet migrated and still runs legacy VPWS:

- At this stage, PE1 keeps running legacy VPWS service with PE3.
- The legacy VPWS to EVPN-VPWS migration then continues to remaining PE nodes. The legacy VPWS and EVPN-VPWS dual-stack coexist in the core for a given L2 Attachment Circuit (AC).

After another year, the user plans to upgrade the PE3 device.

- PE3 is now enabled with EVPN-VPWS service.
- All the PE devices are replaced with EVPN-VPWS services in the network.
- The user plans to retain both legacy and an EVPN-VPWS related configuration on PE1 and PE2 nodes.
- If there are any issues in the network, the user can roll back the migration. After the rollback, the migration to VPWS at node PE2, then PE1 and PE2, will revert to the legacy VPWS between them

Configure EVPN Seamless Integration with Legacy VPWS

To enable the feature, use the **vpws-seamless-integration** command.

Configuration Example

The following example shows how to migrate each PE at a time. In this example, the following Customer Edge (CE) IDs are used:

- PE1 is connected to CE1 and CE3.
- PE2 is connected to CE2.

- PE3 is connected to CE4.

For legacy VPWS configuration, perform the following taks:

1. Configure a cross-connect (xconnect) group for VPWS.
2. Configure a name for xconnect in the mp2mp mode.
3. Configure BGP autodiscovery.
4. Enable BGP signaling.
5. Configure the local CE ID.
6. Configure an interface with the remote CE ID.

The VPWS cross-connect is established between the local and remote CEs.

For migrating the PEs from legacy VPWS to EVPN-VPWS, perform the following taks:

1. In the existing VPWS cross-connect, enable the VPWS seamless integration on the local CE.
2. Configure the interface used in VPWS configuration with the remote CE ID.
3. Configure a cross-connect (xconnect) group for EVPN-VPWS.
4. Configure a name for xconnect in the p2p mode.
5. Assign the interface used in VPWS configuration.
6. Enable EVPN-VPWS on the p2p xconnect.

EVPN-VPWS service is established between the local and remote CEs.

Migration of PE1

In this example, both legacy VPWS and EVPN-VPWS coexist on PE1.

```
/* VPWS configuration on PE1 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# exit
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.2 remote-ce-id 4
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
```

```
/* Migrate VPWS to EVPN-VPWS on PE1 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
```

```
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# vpws-seamless-integration
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# root

Router(config)# l2vpn xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn1
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 service 5
Router(config-l2vpn-xc-p2p-pw)# commit


/* VPWS configuration on PE2 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit


/* VPWS configuration on PE3 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 4
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.2 remote-ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
```

Verification

As PE2 and PE3 are not migrated to EVPN-VPWS, legacy VPWS continues to run between the PE devices. The following show output indicates that only legacy VPWS is up and EVPN-VPWS is down on BE1.1.

```
Router# show l2vpn xconnect

Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed,
        LU = Local Up, RU = Remote Up, CO = Connected, (SI) = Seamless Inactive

XConnect                   Segment 1                       Segment 2
Group        Name       ST   Description            ST       Description            ST
-------------------------  -------------------------   ------------------------------
evpn-vpws    evpn1      DN   BE1.1                  UP       EVPN 4,5,24004          DN
-------------------------------------------------------------------------------------
legacy-vpws  vpws1      UP   BE1.1                  UP       192.168.0.4     534296  UP
-------------------------------------------------------------------------------------
legacy-vpws  vpws1      UP   BE1.2                  UP       192.168.12.110  685694  UP
-------------------------------------------------------------------------------------
```

Migration of PE1 and PE2

In this example, both legacy VPWS and EVPN-VPWS coexist on PE1. PE2 is migrated to EVPN-VPWS.

```
/* VPWS configuration on PE1 */
Router# configure
```

```
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# exit
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.2 remote-ce-id 4
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit

/* Migrate VPWS to EVPN-VPWS on PE1 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# vpws-seamless-integration
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# root

Router(config)# l2vpn xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn1
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 service 5
Router(config-l2vpn-xc-p2p-pw)# commit


/* Migrate VPWS to EVPN-VPWS on PE2 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# vpws-seamless-integration
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# root

Router(config)# l2vpn xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn1
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 service 5
Router(config-l2vpn-xc-p2p-pw)# commit
```

Verification

After the migration, legacy VPWS and EVPN-VPWS coexist on PE1. PE2 is migrated to EVPN-VPWS and PE3 runs with legacy VPWS.

EVPN-VPWS service runs between PE1 and PE2.

Legacy VPWS service runs between PE1 and PE3.

The following example shows that EVPN-VPWS is up on BE1.1. The legacy VPWS is also advertised on BE1.1 with the status as Standby (**SB(SI)**).

```
Router# show l2vpn xconnect

Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed,
        LU = Local Up, RU = Remote Up, CO = Connected, (SI) = Seamless Inactive

XConnect                   Segment 1                       Segment 2
Group        Name       ST   Description            ST       Description            ST
-------------------------  -------------------------   ------------------------------
evpn-vpws    evpn1      UP   BE1.1                  UP       EVPN 4,5,24004          UP
-------------------------------------------------------------------------------------
legacy-vpws  vpws1      DN   BE1.1                  SB(SI)   192.168.0.4     534296  UP
-------------------------------------------------------------------------------------
legacy-vpws  vpws1      UP   BE1.2                  UP       192.168.12.110  685694  UP
-------------------------------------------------------------------------------------
```

Use the **show l2vpn forwarding interface** *interface-type interface-path-id* **detail location** *node-id* command to identify whether EVPN-VPWS or VPWS is used for forwarding the traffic.

In this example, **evi: 1** indicates that EVPN-VPWS is used for forwarding the traffic.

```
Router# show l2vpn forwarding interface Bundle-Ether1.1 detail location 0/2/CPU0
Wed Apr 28 09:08:37.512 EDT
Local interface: Bundle-Ether1.1, Xconnect id: 0x800001, Status: up
  Segment 1
    AC, Bundle-Ether1.1, status: Bound
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
  Segment 2
    MPLS, Destination address: 192.168.0.4, evi: 4, ac-id: 5, status: Bound
Pseudowire label: 24001
    Control word enabled
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
```

In this example, **pw-id: 1** indicates that VPWS is used for forwarding the traffic.

```
Router# show l2vpn forwarding interface interface Bundle-Ether1.1  detail location 0/2/CPU0
Wed Apr 28 09:09:45.204 EDT
Local interface: Bundle-Ether1.1, Xconnect id: 0x800001, Status: up
  Segment 1
    AC, Bundle-Ether1.1, status: Bound
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
  Segment 2
    MPLS, Destination address: 192.168.0.4, pw-id: 1, status: Bound
Pseudowire label: 24000
    Control word disabled
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
```

Use the **l2vpn logging pseudowire** command to track the migration of AC from one PW to another.

```
Router(config)# l2vpn logging pseudowire
RP/0/0/CPU0:Jan 18 15:35:15.607 EST:
l2vpn_mgr[1234]: %L2-EVPN-5-VPWS_SEAMLESS_INTEGRATION_STATE_CHANGE :
```

```
GigabitEthernet0/2/0/8.1 - Active XC is now service-1:evpn-vpws-1, standby XC is
service-1:legacy-vpws-1
```

Migration of PE1, PE2, and PE3

In this example, both legacy VPWS and EVPN-VPWS coexist on PE1. PE2 and PE3 are migrated to EVPN-VPWS.

```
/* VPWS configuration on PE1 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# exit
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.2 remote-ce-id 4
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit

/* Migrate VPWS to EVPN-VPWS on PE1 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 1
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# vpws-seamless-integration
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.1 remote-ce-id 2
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# exit
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# vpws-seamless-integration
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.2 remote-ce-id 4
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# root

Router(config)# l2vpn xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn1
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 service 5
Router(config-l2vpn-xc-p2p-pw)# commit
Router(config-l2vpn-xc-p2p-pw)# root

Router(config)# l2vpn xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn2
Router(config-l2vpn-xc-p2p-pw)# exit
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 service 7
Router(config-l2vpn-xc-p2p-pw)# commit



/* Migrate VPWS to EVPN-VPWS on PE3 */
Router# configure
Router(config)# l2vpn xconnect group legacy-vpws
Router(config-l2vpn-xc)# mp2mp vpws1
Router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
Router(config-l2vpn-xc-mp2mp-ad)# signaling-protocol bgp
Router(config-l2vpn-xc-mp2mp-ad-sig)# ce-id 4
```

```
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# vpws-seamless-integration
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# interface Bundle-Ether1.2 remote-ce-id 3
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# commit
Router(config-l2vpn-xc-mp2mp-ad-sig-ce)# root

Router(config)# l2vpn xconnect group evpn-vpws
Router(config-l2vpn-xc)# p2p evpn2
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 4 service 7
Router(config-l2vpn-xc-p2p-pw)# commit
```

Verification

After migration, all the PE devices forward traffic between them using EVPN-VPWS.

The following example shows that EVPN-VPWS is up and legacy VPWS is down.

```
Router# show l2vpn xconnect

Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed,
        LU = Local Up, RU = Remote Up, CO = Connected, (SI) = Seamless Inactive

XConnect                   Segment 1                       Segment 2
Group        Name       ST   Description            ST       Description            ST
-------------------------  -----------------------------   ------------------------------
evpn-vpws    evpn1      UP   BE1.1                  UP       EVPN 4,5,24004          UP
-------------------------------------------------------------------------------------------
legacy-vpws  vpws1      DN   BE1.1                  UP       192.168.0.4     534296  DN
-------------------------------------------------------------------------------------------
evpn-vpws    evpn2      UP   BE1.2                  UP       EVPN 4,7,24008          UP
-------------------------------------------------------------------------------------------
legacy-vpws  vpws1      DN   BE1.2                  UP       192.168.12.110  685694  DN
-------------------------------------------------------------------------------------------
```

TLDP PW to EVPN-VPWS Migration

Similar to migrating VPWS to EVPN, you can also migrate Targeted Label Distribution Protocol (TLDP) PW to EVPN-VPWS on all the PE routers incrementally.

You can perform this task on all the PE routers incrementaly. The following configuration example shows the TLDP PW to EVPN-VPWS migration on PE1:

```
Router# configure
Router(config)# l2vpn xconnect group 1
Router(config-l2vpn-xc)# p2p p1
Router(config-l2vpn-xc-p2p)# interface BE1.1
Router(config-l2vpn-xc-p2p)# neighbor 10.0.0.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# exit
Router(config-l2vpn-xc-p2p)# vpws-seamless-integration
```

Private Line Emulation over EVPN-VPWS Single Homed

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Private Line Emulation over EVPN-VPWS Single Homed	Release 7.11.1	Introduced in this release on: Cisco 8011-2X2XP4L PLE Service Endpoint Router. You can now configure EVPN VPWS to carry the client traffic from ports like FC, OTN, SDH, SONET, or Ethernet and forward the traffic to the core network by using Private Line Emulation (PLE). PLE emulates the switching capabilities of FC, OTN, SDH, SONET, or Ethernet ports without needing a dedicated equipment and allows interconnecting optical networks with Ethernet networks. This feature introduces the **port-mode** command. This release introduces new and modified YANG data models for PLE. For the list of supported data models, see *Supported Yang Data Models for PLE*. You can access these data models from the Github repository.

PLE service is a mechanism that allows the transparent transfer of packets from different port modes over MPLS networks.

PLE client traffic is carried on EVPN-VPWS single homed service. The PLE endpoints establish a BGP session to exchange EVPN route information. The pseudowire channel is set up between the endpoints when the L2VPN cross-connect is set up between PLE client, represented as Circuit Emulation (CEM) interface, and the remote node.

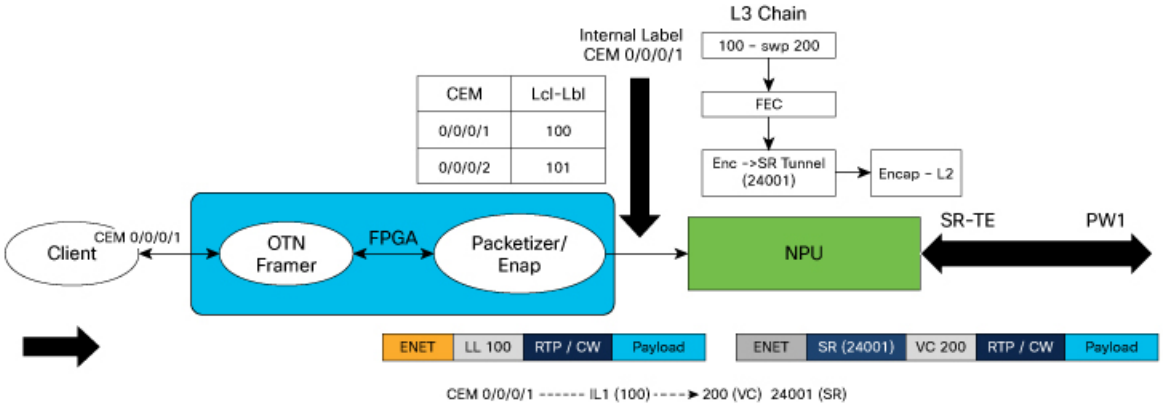
CEM helps PLE endpoints to provide native client interfaces. CEM service is a method through which data can be transmitted over Ethernet or MPLS networks. CEM over a packet carries circuits over Packet Switched Network (PSN) placing the client bitstreams into packet payload with appropriate pseudowire emulation headers.

PLE client traffic is encapsulated by PLE initiator and is carried over EVPN-VPWS L2 service running on segment routing or MPLS tunnels. PLE terminator node extracts the bitstreams from the EVPN packets and places them to the PLE client interface as defined by the client attribute and CEM profile. The traffic flow between the client and core networks happens with label imposition and disposition.

PLE Forwarding Flow – Imposition

Imposition is the process of adding an MPLS label to a data packet. A PE router forwards traffic from a client interface by adding an MPLS label to the packet upon entering an MPLS network. When PLE forwards traffic from client to core network, label imposition is used to forward the packets.

Figure 6: PLE Forwarding Flow – Imposition



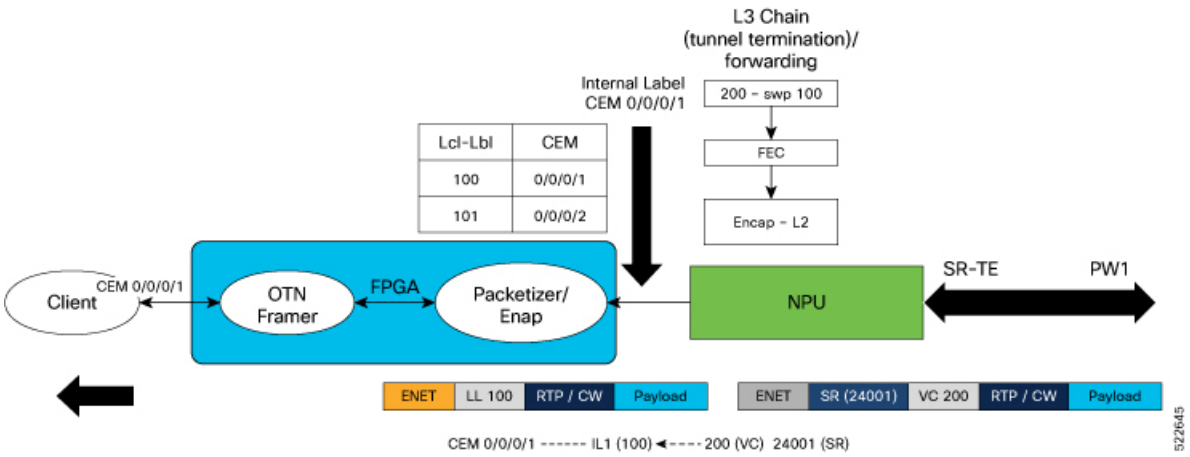
In the diagram, traffic from client may be of any port mode like FC, OTN, SDH, SONET, or Ethernet. Field Programmable Gate Array (FPGA) acts as a forwarding block. FPGA sends the traffic from the client towards NPU with an assigned internal local label.

- In this example, the traffic from client flows through CEM interface. The internal local label 100 is added to the CEM interface 0/0/0/1 in the FPGA.
- NPU receives traffic with assigned internal local label from FPGA and in the forwarding L3 chain, replaces the internal local label 100 with Virtual Circuit (VC) label 200. VC label is also known as the pseudowire (PW) label.
- The traffic is then forwarded towards core network using the transport label 24001.

PLE Forwarding Flow – Disposition

Disposition is the process of removing an MPLS label from a data packet. A PE router receives an MPLS packet, makes a forwarding decision based on the MPLS label, removes the label, and sends the traffic to the client. When PLE forwards traffic from core to client network, label disposition is used to forward the packets.

Figure 7: PLE Forwarding Flow – Disposition



In the diagram, NPU receives traffic with VC label.

- NPU determines the outgoing interface for the traffic, based on the VC label allocation.
- The VC label 200 is replaced with the internal local label 100 and sent to FPGA.
- In the FPGA, the internal local label is mapped to CEM interface 0/0/0/1 and traffic is forwarded to the client through the CEM interface.

PLE Transport Mechanism

You can configure circuit-style segment routing to transport PLE client traffic over the networks. Circuit-style SR-TE supports the following:

- Co-router bidirectional paths
- Guaranteed latency
- End-to-end path protection
- Guaranteed bandwidth

The circuit-style SR-TE policies are configured statically as preferred path within a pseudowire class. An SR-TE policy is associated per pseudowire by assigning corresponding pseudowire class to working or protected pseudowires.

For more information on SR-TE policies, see the *Configure SR-TE Policies* section in the *Segment Routing Configuration Guide for Cisco 8000 Series Routers, IOS XR.*

Supported Hardware for PLE

PLE is supported on Cisco 8011-2X2XP4L PLE Service Endpoint Router with SFP+ optical transceivers and supports the following port mode options:

- Ethernet – 10GE
- Fiber channel (FC) – 1G, 2G, 4G, 8G, 16G, and 32G
- Optical Transport Network (OTN) – OTU2 and OTU2e

- Synchronous Digital Hierarchy (SDH) – STM16 and STM64
- Synchronous Optical Networking (SONET) – OC48 and OC192

Restrictions for PLE over EVPN VPWS

Note These following restrictions are applicable only to Cisco 8011-2X2XP4L PLE Service Endpoint Router for IOS XR Release 7.11.1.

- Load balancing is not supported for PLE traffic in the core, because PLE does not work with ECMP or core bundle having more than one member link.
- Software offloading is supported only on SR-TE performance monitoring and hence Fast Reroute (FRR) convergence is not possible.
- PLE circuit over SR-TE tunnel with deep label stack is not supported, as this may lead to the circuit being down. For more information on label stacking, see *MPLS Configuration Guide for Cisco 8000 Series Routers, IOS XR*.

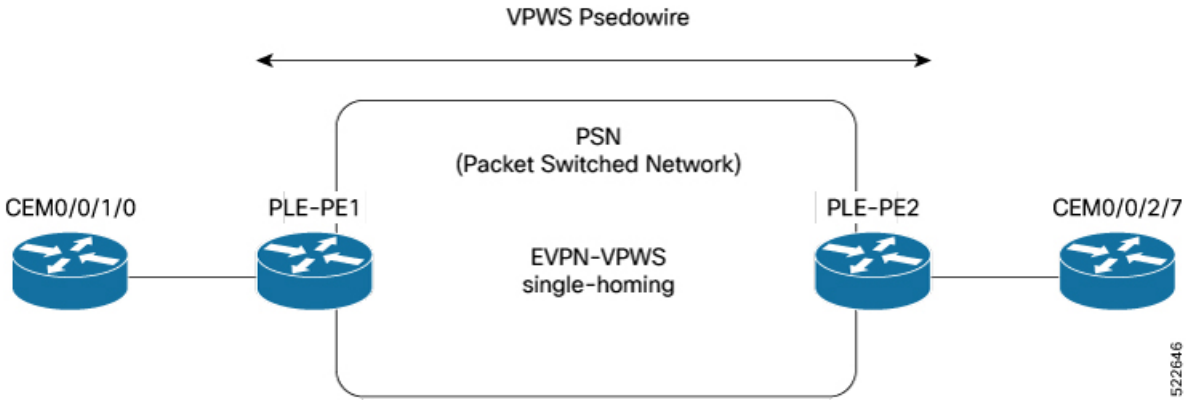
Configure PLE over EVPN VPWS

Prerequisites

- Install all the mandatory Cisco RPMS like RSVP for MPLS-TE. For more information, see the *Implementing RSVP for MPLS-TE* section in the *MPLS Configuration Guide for Cisco 8000 Series Routers, IOS XR*.
- Ensure that the clocks between the routers in the network is synchronized with Synchronous Ethernet (SyncE) or Precision Time Protocol (PTP), to avoid drop in the data traffic.
- Core interface bandwidth must be higher than the access interface. For example, when traffic from CE is 10G, it becomes 12.5G when it reaches the core. Hence, the core interface bandwidth must be at least 25G.

Topology

Figure 8: PLE over EVPN VPWS



In this topology, CEM interfaces are connected to PLE interfaces. The PLE interfaces, PE1 and PE2, are connected through EVPN-VPWS single homing. The PLE interface can be: Ethernet, OTN, FC, or SONET/SDH.

Configuration Example

Perform the following tasks to configure EVPN-VPWS over SR-TE policy with explicit path. For more information on SR-TE policies, see the *Configure SR-TE Policies* section in the *Segment Routing Configuration Guide for Cisco 8000 Series Routers, IOS XR*.

1. Enable Frequency Synchronization to synchronize the clock between the PE routers.
2. Bring up the Optics Controller in CEM Packet Mode, based on the port mode type.
3. Configure Access and Core Interfaces.
4. Configure Loopback Interface to establish BGP-EVPN neighborship.
5. Configure IS-IS IGP to advertise the loopback and core interfaces.
6. Configure Performance Measurement to enable liveness monitoring of SR policy.
7. Configure Segment Routing Traffic Engineering Tunnels with circuit-styled SR-TE tunnels and explicit path.
8. Configure BGP EVPN Neighbor Session to exchange EVPN route information.
9. Configure EVPN VPWS with pseudowire class (PW) and cross-connect (xconnect) service to carry the PLE client traffic.
10. Configure QoS Policy on CEM Interface to manage congestion on PLE client traffic.

Enable Frequency Synchronization

Synchronize the clocks between PE1 and PE2.

```
/* Enable Frequency Synchronization on PLE-PE1 */
```

Prerequisites: SyncE or PTP must be UP.

```
Router(config)# frequency synchronization
Router(config-freqsync)# quality itu-t option 1
Router(config-freqsync)# exit
Router(config)# interface TwentyFiveGigE0/0/0/24
Router(config-if)# frequency synchronization
Router(config-if-freqsync)# quality transmit exact itu-t option 1 PRC
!
```

(Use the **show frequency synchronization interfaces** command to verify that the clock is transmitted.)

```
/* Enable Frequency Synchronization on PLE-PE2 */
Router(config)# frequency synchronization
Router(config-freqsync)# quality itu-t option 1
Router(config-freqsync)# exit
Router(config)# interface TwentyFiveGigE0/0/0/32
Router(config-if)# frequency synchronization
Router(config-if-freqsync)# selection input
Router(config-if-freqsync)# priority 1
Router(config-if-freqsync)# wait-to-restore 0
!
```

(Use the **show frequency synchronization selection** command to verify if PLE-PE2 is LOCKED to PLE-PE1's clock.)

Bring up the Optics Controller in CEM Packet Mode

Configure the optics controller and port mode. The examples show port mode configuration for all the types of port modes. Use the relevant command according to the port mode type of the PLE interface.

```
/* Bring up the optics controller in CEM packet mode with appropriate speed on PLE-PE1 */
```

Ethernet:

```
Router(config)# controller Optics0/0/1/0
Router(config-Optics)# port-mode Ethernet framing cem-packetize rate 10GE
!
Router(config)# controller Optics0/0/1/5
Router(config-Optics)# port-mode Ethernet framing cem-packetize rate 1GE
!
```

OTN:

```
Router(config)# controller Optics0/0/2/0
Router(config-Optics)# port-mode otn framing cem-packetize rate otu2
!
Router(config)# controller Optics0/0/2/0
Router(config-Optics)# port-mode otn framing cem-packetize rate otu2e
!
```

Fiber Channel:

```
Router(config)# controller Optics0/0/1/6
Router(config-Optics)# port-mode FC framing cem-packetize rate FC1
```

Note Port mode FC32 is supported only on the even ports (Port 0, 2, 4, and 6) of the MPA.

SONET/SDH:

```
Router(config)# controller optics 0/0/2/4
Router(config-Optics)# port-mode sonet framing cem-packetize rate OC48
!
Router(config)# controller optics 0/0/2/5
Router(config-Optics)# port-mode sdh framing cem-packetize rate STM16
!
```

```
/* Bring up the optics controller in CEM packet mode with appropriate speed on PLE-PE2 */
```

Ethernet:

```
Router(config)# controller Optics0/0/2/7
Router(config-Optics)# port-mode Ethernet framing cem-packetize rate 10GE
!

Router(config)# controller Optics0/0/1/5
Router(config-Optics)# port-mode Ethernet framing cem-packetize rate 1GE
```

OTN:

```
Router(config)# controller Optics0/0/2/0
Router(config-Optics)# port-mode otn framing cem-packetize rate otu2
!

Router(config)# controller Optics0/0/2/0
Router(config-Optics)# port-mode otn framing cem-packetize rate otu2e
!
```

Fiber Channel:

```
Router(config)# controller Optics0/0/1/6
Router(config-Optics)# port-mode FC framing cem-packetize rate FC1
```

Note Port mode FC32 is supported only on the even ports (Port 0, 2, 4, and 6) of the MPA.

SONET/SDH:

```
Router(config)# controller optics 0/0/2/4
Router(config-Optics)# port-mode sonet framing cem-packetize rate OC48
!
Router(config)# controller optics 0/0/2/5
Router(config-Optics)# port-mode sdh framing cem-packetize rate STM16
!
```

Configure Access and Core Interfaces

Configure the access interface for the client and then the core interface.

```
/* Configure the access and core interfaces on PLE-PE1 */
```

Access interface: Repeat this for each port mode configuration.

```
Router(config)# interface CEM0/0/1/0
Router(config-if)# l2transport
!
```

Core interface:

```
Router(config)# interface TwentyFiveGigE0/0/0/24
Router(config-if)# ipv4 address 14.1.0.1 255.255.255.252
!
```

```
/* Configure the access and core interfaces on PLE-PE2 */
```

Access interface: Repeat this for each port mode configuration.

```
Router(config)# interface CEM0/0/2/7
Router(config-if)# l2transport
!
```

Core interface:

```
Router(config)# interface TwentyFiveGigE0/0/0/32
Router(config-if)# ipv4 address 14.1.0.2 255.255.255.252
!
```

Configure Loopback Interface

Configure loopback interface to establish BGP-EVPN neighborship.

```
/* Configure loopback interface on PLE-PE1 */
```

```
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 1.1.1.1 255.255.255.255
!
```

```
/* Configure loopback interface on PLE-PE2 */
```

```
Router(config)# interface Loopback0
Router(config-if)# ipv4 address 1.1.1.4 255.255.255.255
!
```

Configure IS-IS IGP

Configure IS-IS IGP to advertise the configured loopback and core interfaces.

Note You cannot configure Topology-Independent Loop-Free Alternate (TI-LFA) on the links used by circuit-styled SR-TE tunnel. The adjacency SID label is unprotected for circuit-styled SR-TE, which does not support TI-LFA.

```
/* Configure IS-IS IGP on PLE-PE1 */
```

```
Router(config)# router isis core
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.0000.0000.0000.0001.00
Router(config-isis)# nsr
Router(config-isis)# nsf cisco
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# segment-routing mpls sr-prefer
Router(config-isis-af)# segment-routing bundle-member-adj-sid
```

```
Router(config-isis-af)# commit
Router(config-isis-af)# exit

Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# prefix-sid index 1
Router(config-isis-if-af)# exit
!
Router(config-isis)# interface TwentyFiveGigE0/0/0/24
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# adjacency-sid absolute 28121 >>>>> Adjacency-SID must be
unprotected for circuit-styled SR-TE
Router(config-isis-if-af)# commit
Router(config-isis-if-af)# exit
!
!
/* Configure IS-IS IGP on PLE-PE2 */


Router(config)# router isis core
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.0000.0000.0000.0004.00
Router(config-isis)# nsr
Router(config-isis)# nsf cisco
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# segment-routing mpls sr-prefer
Router(config-isis-af)# segment-routing bundle-member-adj-sid
Router(config-isis-af)# commit
Router(config-isis-af)# exit

Router(config-isis)# interface Loopback0
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# prefix-sid index 4
Router(config-isis-if-af)# exit
!
!
Router(config-isis)# interface TwentyFiveGigE0/0/0/32
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# adjacency-sid absolute 28211 >>>>> Adjacency-SID must be
unprotected for circuit-styled SR-TE
Router(config-isis-if-af)# commit
Router(config-isis-if-af)# exit
!
!
```

Configure Performance Measurement

Configure the performance measurement to enable the liveness monitoring of the SR policy.

```
/* Configure performance measurement on PLE-PE1 */


Router(config)# performance-measurement
Router(config-perf-meas)# liveness-profile sr-policy name RED
Router(config-pm-ld-srpolicy)# probe
Router(config-pm-ld-srpolicy-probe)# measurement-mode loopback
Router(config-pm-ld-srpolicy-probe)# burst-interval 3000
```

```
Router(config-pm-ld-srpolicy-probe)# exit
Router(config-pm-ld-srpolicy)# exit

Router(config-perf-meas)# liveness-profile sr-policy name BLUE
Router(config-pm-ld-srpolicy)# probe
Router(config-pm-ld-srpolicy-probe)# measurement-mode loopback
Router(config-pm-ld-srpolicy-probe)# burst-interval 30

/* Configure performance measurement on PLE-PE2 */


Router(config)# performance-measurement
Router(config-perf-meas)# liveness-profile sr-policy name RED
Router(config-pm-ld-srpolicy)# probe
Router(config-pm-ld-srpolicy-probe)# measurement-mode loopback
Router(config-pm-ld-srpolicy-probe)# burst-interval 3000
Router(config-pm-ld-srpolicy-probe)# exit
Router(config-pm-ld-srpolicy)# exit

Router(config-perf-meas)# liveness-profile sr-policy name BLUE
Router(config-pm-ld-srpolicy)# probe
Router(config-pm-ld-srpolicy-probe)# measurement-mode loopback
Router(config-pm-ld-srpolicy-probe)# burst-interval 30
```

Configure Segment Routing Traffic Engineering Tunnels

Configure circuit-styled SR-TE tunnels. SR-TE is supported only with explicit path specified by adjacency SID labels. The adjacency SID labels must be unprotected for circuit-styled SR-TE. This example shows configuration of explicit path between PE1 and PE2.

```
/* Configure segment routing traffic engineering tunnels on PLE-PE1 */


Router(config)# segment-routing
Router(config-sr)# global-block 80000 111999
Router(config-sr)# local-block 25000 28999
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list pe1-pe2-forward-path
Router(config-sr-te-sl)# index 1 mpls label 28121
Router(config-sr-te-sl)# exit

Router(config-sr-te)# segment-list pe1-pe2-reverse-path
Router(config-sr-te-sl)# index 1 mpls label 28211
Router(config-sr-te-sl)# exit

Router(config-sr-te)# policy pe1-pe2-circuit-styled-srte
Router(config-sr-te-policy)# color 10 end-point ipv4 1.1.1.4
Router(config-sr-te-policy)# path-protection
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 10
Router(config-sr-te-policy-path-pref)# explicit segment-list pe1-pe2-forward-path >>>>>
Explicit path
Router(config-sr-te-policy-path-pref)# reverse-path segment-list pe1-pe2-reverse-path
!
!
!
Router(config)# performance-measurement
Router(config-perf-meas)# liveness-detection
Router(config-perf-meas)# liveness-profile backup name RED
Router(config-perf-meas)# liveness-profile name BLUE

/* Configure segment routing traffic engineering tunnels on PLE-PE2 */
```

```
Router(config)# segment-routing
Router(config-sr)# global-block 80000 111999
Router(config-sr)# local-block 25000 28999
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list pe1-pe2-forward-path
Router(config-sr-te-sl)# index 1 mpls label 28211
Router(config-sr-te-sl)# exit

Router(config-sr-te)# segment-list pe1-pe2-reverse-path
Router(config-sr-te-sl)# index 1 mpls label 28121
Router(config-sr-te-sl)# exit

Router(config-sr-te)# policy pe1-pe2-circuit-styled-srte
Router(config-sr-te-policy)# color 10 end-point ipv4 1.1.1.1
Router(config-sr-te-policy)# path-protection
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 10
Router(config-sr-te-policy-path-pref)# explicit segment-list pe1-pe2-forward-path >>>>>
Explicit path
Router(config-sr-te-policy-path-pref)# reverse-path segment-list pe1-pe2-reverse-path
!
!
!
Router(config)# performance-measurement
Router(config-perf-meas)# liveness-detection
Router(config-perf-meas)# liveness-profile backup name RED
Router(config-perf-meas)# liveness-profile name BLUE
```

Configure BGP EVPN Neighbor Session

Configure L2VPN EVPN address family under BGP to establish a BGP-EVPN neighbor session.

```
/* Configure BGP EVPN neighbor session on PLE-PE1 */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 1.1.1.1
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 1.1.1.4
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# exit
Router(config-bgp)# graceful-restart
Router(config-bgp)# address-family l2vpn evpn

/* Configure BGP EVPN neighbor session on PLE-PE2 */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 1.1.1.4
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp-af)# exit
Router(config-bgp)# neighbor 1.1.1.1
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# exit
```

```
Router(config-bgp)# graceful-restart
Router(config-bgp)# address-family l2vpn evpn
```

Configure EVPN VPWS

Configure EVPN VPWS with PW class and xconnect service to carry the PLE client traffic.

```
/* Configure EVPN VPWS on PLE-PE1 */


Router(config)# l2vpn
Router((config-l2vpn)# pw-class pw-cs-srte
Router((config-l2vpn-pwc)# encapsulation mpls
Router((config-l2vpn-pwc-mpls)# preferred-path sr-te policy srte_c_10_ep_1.1.1.6
!
!
Router(config)# xconnect group evpn_vpws
Router(config)# p2p p1
Router(config)# interface CEM0/0/1/0
Router(config)# neighbor evpn evi 10 target 1 source 2
Router(config)# pw-class pw-cs-srte
```

```
/* Configure EVPN VPWS on PLE-PE2 */


Router(config)# l2vpn
Router((config-l2vpn)# pw-class pw-cs-srte
Router((config-l2vpn-pwc)# encapsulation mpls
Router((config-l2vpn-pwc-mpls)# preferred-path sr-te policy srte_c_10_ep_1.1.1.1
!
!
Router(config)# xconnect group evpn_vpws
Router(config)# p2p p1
Router(config)# interface CEM0/0/2/7
Router(config)# neighbor evpn evi 10 target 1 source 2
Router(config)# pw-class pw-cs-srte
```

Configure QoS Policy on CEM Interface

Configure QoS policy to manage congestion on PLE client traffic. In QoS for PLE, you can mark the MPLS experimental with only the topmost label and set the traffic class with only the default class.

```
/* Configure QoS policy on PLE-PE1 */
```

Access Interface Configuration

```
Router(config)# policy-map ple-policy
Router(config-pmap)# class class-default
Router(config-pmap-c)# set mpls experimental topmost 7
Router(config-pmap-c)# set traffic-class 2
Router(config-pmap-c)# end-policy-map
!

Router(config)# interface CEM0/0/1/0
Router(config-if)# l2transport
Router(config-if)# service-policy input ple-policy
!
!
```

Core Interface Configuration

```
Router(config)# class-map match-any tc2
Router(config-cmap)# match traffic-class 2
```

```
Router(config-cmap)# end-class-map
!

Router(config)# policy-map core
Router(config-pmap)# class tc2
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# shape average percent 100
Router(config-pmap-c)# end-policy-map
!

Router(config)# interface TwentyFiveGigE0/0/0/24
Router(config-if)# mtu 9200
Router(config-if)# service-policy output core
Router(config-if)# ipv4 address 13.30.1.1 255.255.255.252

/* Configure QoS policy on PLE-PE2 */
```

Access Interface Configuration

```
Router(config)# policy-map ple-policy
Router(config-pmap)# class class-default
Router(config-pmap-c)# set mpls experimental topmost 7
Router(config-pmap-c)# set traffic-class 2
Router(config-pmap-c)# end-policy-map
!
Router(config)# interface CEM0/0/2/7
Router(config-if)# l2transport
Router(config-if)# service-policy input ple-policy
```

Core Interface Configuration

```
Router(config)# class-map match-any tc2
Router(config-cmap)# match traffic-class 2
Router(config-cmap)# end-class-map
!

Router(config)# policy-map core
Router(config-pmap)# class tc2
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# shape average percent 100
Router(config-pmap-c)# end-policy-map
!

Router(config)# interface TwentyFiveGigE0/0/0/32
Router(config-if)# mtu 9200
Router(config-if)# service-policy output core
Router(config-if)# ipv4 address 46.10.1.2 255.255.255.252
```

Verification

Use the following show commands to view the configuration.

Verify the IS-IS configuration.

```
Router# show isis neighbors
Fri Nov 12 09:04:13.638 UTC

IS-IS core neighbors:
System Id      Interface        SNPA           State Holdtime Type IETF-NSF
PLE-Core-PE2   TF0/0/0/24       *PtoP*         Up    28       L2   Capable

Total neighbor count: 1
```

```
Router# show isis segment-routing label table

Fri Nov 12 09:25:18.488 UTC

IS-IS core IS Label Table
Label         Prefix                 Interface
----------    ----------------       ---------
16001         1.1.1.1/32             Loopback0
16004         1.1.1.4/32

Router# show mpls forwarding prefix 1.1.1.4/32
Fri Nov 12 09:25:54.898 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16004  Pop         SR Pfx (idx 4)     TF0/0/0/24   14.1.0.2        104332
```

Verify the performance measurement.

```
Router# show performance-measurement sr-policy color 203
Mon Mar 14 17:54:32.403 IST

------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------

SR Policy name: srte_c_203_ep_1.1.1.1
Color : 203
Endpoint : 1.1.1.1
Number of candidate-paths : 1

Candidate-Path:
Instance : 8
Preference : 10
Protocol-origin : Configured
Discriminator : 10
Profile Keys:
Profile name : BLUE
Profile type : SR Policy Liveness Detection
Source address : 1.1.1.6
Number of segment-lists : 1
Liveness Detection: Enabled
Session State: Up
Last State Change Timestamp: Mar 14 2022 17:53:45.207
Missed count: 0

------------------------------------------------------------------
0/0/CPU0
------------------------------------------------------------------
```

Verify SR-TE configuration.

```
Router# show segment-routing traffic-eng policy color 10 tabular
Fri Nov 12 09:15:57.366 UTC

 Color                              Endpoint  Admin   Oper                Binding
                                              State   State                 SID
------ ------------------------------------ ------ ------ ----------------------
    10                               1.1.1.4     up     up                 24010
```

Verify BGP EVPN neighbor session configuration.

```
Router# show bgp l2vpn evpn neighbors brief
Fri Nov 12 09:10:22.999 UTC
```

```
Neighbor        Spk    AS Description                          Up/Down  NBRState
1.1.1.4           0   100                                      15:51:52 Established
```

Verify EVPN VPWS configuration.

```
Router# show l2vpn xconnect
Fri Nov 12 09:02:44.982 UTC
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed,
        LU = Local Up, RU = Remote Up, CO = Connected, (SI) = Seamless Inactive

XConnect                    Segment 1                       Segment 2
Group      Name       ST    Description            ST       Description            ST
------------------------------------------------------------------------------------
evpn_vpws  p1         UP    CE0/0/1/0              UP       EVPN 10,1,24012        UP
------------------------------------------------------------------------------------
```

Verify QoS policy configuration.

The following show command displays information about interfaces on which the policy maps are applied.

```
Router# show policy-map targets
Thu Jun 16 21:47:31.407 IST
1) Policymap: ple-p1 Type: qos
Targets (applied as main policy):
CEM0/0/1/0 input
Total targets: 1

Targets (applied as child policy):
Total targets: 0

2) Policymap: core Type: qos
Targets (applied as main policy):
TwentyFiveGigE0/0/0/24
Total targets: 1

Targets (applied as child policy):
Total targets: 0
```

Use the following show command to view the core interface information and to verify the traffic class (TC) mapping in CEM interface.

```
Router# Show policy-map interface TwentyFiveGigE0/0/0/24
Thu Jun 16 21:37:52.915 IST
TwentyFiveGigE0/0/0/24 direction input: Service Policy is not installed
TwentyFiveGigE0/0/0/24 output: core
Class tc2
    Classification Statistics    (packets/bytes)          (rate - kbps)
        Matched           :      39654778/42113374236        6816279
        Transmitted       :      39654778/42113374236        6816279
        Total Dropped     :             0/0                  0
    Queueing Statistics
        Queue ID                      : 1370
        Taildropped(packets/bytes)    : 0/0
Class class-default
    Classification Statistics    (packets/bytes)          (rate - kbps)
        Matched           :             0/0                  0
        Transmitted       :             0/0                  0
        Total Dropped     :             0/0                  0
    Queueing Statistics
        Queue ID                      : 1368
        Taildropped(packets/bytes)    : 0/0
Policy Bag Stats time: 1655395669491    [Local Time: 06/16/22 21:37:49:491]
```

Supported Yang Data Models for PLE

The following is the list of new and modified Yang data models supported for PLE. You can access the data models from the Github repository.

Configuration Files - New:

- Cisco-IOS-XR-controller-fc-cfg.yang
- Cisco-IOS-XR-fibrechannelmib-cfg.yang
- Cisco-IOS-XR-interface-cem-cfg.yang
- Cisco-IOS-XR-cem-class-cfg.yang

Configuration Files - Modified:

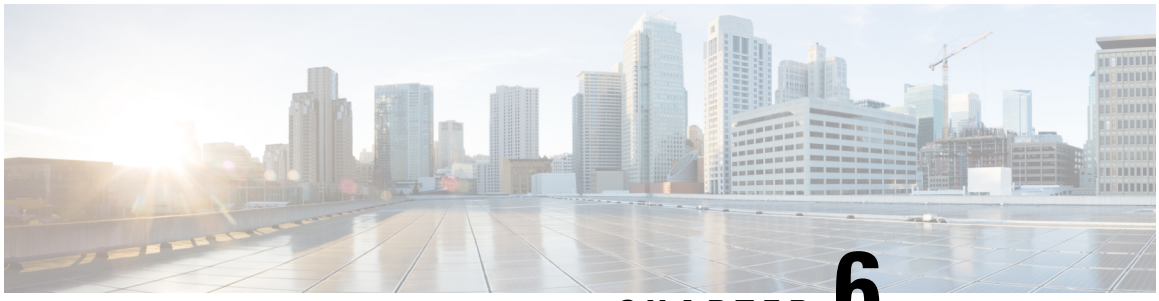
- Cisco-IOS-XR-controller-odu-cfg.yang
- Cisco-IOS-XR-controller-otu-cfg.yang
- Cisco-IOS-XR-controller-sonet-cfg.yang
- Cisco-IOS-XR-drivers-icpe-ethernet-cfg.yang

Operational Files - New:

- Cisco-IOS-XR-controller-fc-oper.yang
- Cisco-IOS-XR-interface-cem-oper.yang

Operational Files - Modified:

- Cisco-IOS-XR-controller-odu-oper.yang
- Cisco-IOS-XR-controller-otu-oper.yang



CHAPTER 6

EVPN Features

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

BUM Ingress Replication for EVPN Single-Homing

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
BUM Ingress Replication for EVPN Single-Homing	Release 7.11.1	You can optimize Broadcast, Unknown Unicast, and Multicast (BUM) traffic by ensuring that traffic that a single-homed device receives is replicated and forwarded to only those CE devices in an EVPN network, if and when they require it. This reduction in the unnecessary forwarding of BUM traffic prevents the flooding of BUM traffic to all devices on the EVPN network. This feature is supported only on Q200-based line cards. This feature is enabled by default.

EVPN BUM ingress replication handles the forwarding of BUM traffic within the network. When the router receives BUM traffic from a particular source, it replicates and forwards that traffic to all the relevant

destinations. EVPN BUM ingress replication involves replicating the BUM traffic at the ingress (source) device and forwarding it to the appropriate egress (destination) devices.

The EVPN protocol uses MAC advertisement routes and control plane signaling to enable routers to selectively forward traffic to the intended recipient devices, preventing flooding the entire EVPN network.

Here's How it Works

1. The ingress router learns MAC addresses associated with BUM traffic and also gathers information about multicast group membership for multicast traffic.
2. Instead of flooding the BUM traffic across all ports in the EVPN, the ingress router replicates the traffic to the specific EVPN instances or Virtual Routing and Forwarding instances (VRFs) where it is needed. This ensures that BUM traffic is forwarded solely to the appropriate destinations.
3. BGP signals and distributes MAC addresses and multicast information to other routers in the EVPN network. Ingress router replicates and transmits the BUM traffic to the designated locations.

Feature Highlights

- EVPN BUM ingress replication optimizes resource allocation by forwarding BUM traffic to necessary EVPN instances or VRFs, minimizing traffic flooding, reducing congestion, and preserving bandwidth. Thereby, reducing the overhead and potential performance issues within a broadcast domain.
- Effective BUM traffic management is vital for sustaining network scalability. EVPN BUM ingress replication guarantees directed BUM traffic distribution, supporting network scaling without overwhelming broadcast or multicast traffic.
- EVPN BUM ingress replication improves network security by selectively duplicating BUM traffic to the designated recipients, thus limiting sensitive traffic exposure to unintended devices and unauthorized access risks.
- In traditional Ethernet networks, broadcast storms occur when broadcast traffic is flooded throughout the network. EVPN BUM ingress replication reduces this risk by forwarding traffic only to the intended devices, preventing broadcast storms and network disruptions.

Split-Horizon Groups for EVPN Single-Homing

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Split-Horizon Groups for EVPN Single-Homing	Release 7.11.1	You can prevent unnecessary BUM traffic flooding and conserve bandwidth for single-homed EVPN scenarios by ensuring that the traffic isn't sent back to the CE device from which it originated. Depending on the type of traffic you need to be forwarded or distributed, you can configure split-horizon group 0 (SG 0), SG 1, or SG 2. This feature is supported only on Q200-based line cards. The feature introduces the **split-horizon group** command.

Split horizon is a method for preventing loops in a network by placing forwarding or flooding restrictions between bridge ports based on group membership. The bridge domain aggregates attachment circuits (ACs) and pseudowires (PWs) in one of three groups called split-horizon groups. When applied to bridge domains, split-horizon refers to the flooding and forwarding behavior between members of a split-horizon group. Bridge domain traffic is either unicast or flooding.

Traffic flooding is performed for broadcast, multicast and unknown unicast destination address. Unicast traffic consists of frames sent to bridge-ports where the destination MAC address is known.

Flooding traffic consists of:

- Unknown unicast destination MAC address frames
- Frames sent to Ethernet multicast addresses, such as Spanning Tree Bridge Protocol Data Units (BPDUs).
- Ethernet broadcast frames (MAC address FF-FF-FF-FF-FF-FF)

Members within certain groups are forbidden to send traffic to each other. Members in different groups can send traffic to each other without restriction. These groups divide the network into segments with unique routes, improving traffic control and reducing packet congestion. This approach enhances network performance and reliability.

The following table describes how frames received on one member of a split-horizon group are treated and if the traffic is forwarded to the other members of the same split-horizon group. It describes the behavior of forwarding and flooding within and between groups as well as the assignment of Bridge Ports (BPs) to groups:

Table 11: Supported Split-Horizon Groups

Split-Horizon Group	Behavior
0	Default AC group. There is no forwarding and flooding restrictions. Forwards and floods traffic within the group and between all groups. By default, all L2 ACs are added to this group by default. You cannot assign L2 ACs manually through the CLI.
1	Default VFI (core) PW group. Forwarding or flooding of traffic is restricted between bridge ports in this group. Forwarding or flooding of traffic to all other groups is allowed. All EVPN EVI virtual ports and VFI PWs are added to this group. You cannot assign VFI PWs manually through the CLI.
2	Optional AC group. Forwarding or flooding of traffic is restricted between bridge ports in this group. Forwarding or flooding traffic to all other groups is allowed. You can manually add ACs, and not VFI PWs using the CLI.

Configuration Example

Perform this task to configure split-horizon groups:

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg
Router(config-l2vpn-bg)# bridge-domain bd
Router(config-l2vpn-bg-bd-ac)# interface HundredGigE 0/0/0/24 <- (split-horizon group 0,
default)
Router(config-l2vpn-bg-bd-ac)# interface interface HundredGigE 0/0/0/24.1
Router(config-l2vpn-bg-bd-evi)# split-horizon group <- (split-horizon group 2)
Router(config-l2vpn-bg-bd-evi)# neighbor 10.0.0.1 pw-id 1
Router(config-l2vpn-bg-bd-pw)# split-horizon group <- (split-horizon group 2)
Router(config-l2vpn-bg-bd-pw)# vfi vf
Router(config-l2vpn-bg-bd-vfi)# neighbor 172.16.0.1 pw-id 10001 <- (split-horizon group 1,
 default)
Router(config-l2vpn-bg-bd-vfi-pw)# exit
Router(config-l2vpn-bg-bd-vfi)# exit
Router(config-l2vpn-bg-bd)# evi 200
Router(config-l2vpn-bg-bd-evi)# commit
```

Running Configuration

This section shows the split-horizon groups running configuration.

```
l2vpn
  bridge group bg
   bridge-domain bd
    interface HundredGigE 0/0/0/24 <- (split-horizon group 0, default)
    interface HundredGigE 0/0/0/24.1
```

```
   !

     split-horizon group <- (split-horizon group 2)
    neighbor 10.0.0.1 pw-id 1
     split-horizon group <- (split-horizon group 2)
    vfi vf
      neighbor 172.16.0.1 pw-id 10001 <- (split-horizon group 1, default)
  !
   evi 200
!
```

Verification

The **show l2vpn bridge-domain detail** command output displays information about bridges, including whether each AC is in the AC split-horizon group or not.

```
Router# show l2vpn bridge-domain detail | i "AC:|Split Horizon|PW:|VFI"
MAC withdraw for Access PW: enabled
Split Horizon Group: none
P2MP PW: disabled
ACs: 2 (2 up), VFIs: 1, PWs: 2 (0 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
AC: HundredGigE 0/0/0/24, state is up
Split Horizon Group: none
AC: HundredGigE 0/0/0/24.1, state is up
Split Horizon Group: enabled
PW: neighbor 10.0.0.1, pw-id 1, state is up ( established )
Split Horizon Group: enabled
List of VFIs:
VFI vf (up)
PW: neighbor 172.16.0.1, pw-id 10001, state is up ( established )
Split Horizon Group: none
```

VRF Leaking for EVPN Single-Homing

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
VRF Leaking for EVPN Single-Homing	Release 7.11.1	We now allow for seamless intercommunication between different VRF instances in an EVPN domain, thus enabling controlled inter-VRF communication and resource-sharing, which is helpful in multi-tenancy environments, data center deployments, and hybrid cloud scenarios. This feature is supported only on Q200-based line cards.

For virtualization, multiple virtual networks are created using VRFs to provide logical separation of network resources, enabling different network domains to have their own isolated virtual networks. Each VRF operates

in isolation with its own routing table, forwarding behavior, and network policies. Devices within a VRF can communicate with each other but are isolated from devices in other VRFs.

However, these isolated domains often need to communicate with each other, such as providing services, sharing resources, centralized management, and monitoring.

In EVPN single-homing, VRF leaking facilitates the controlled exchange of information between different VRF instances within an EVPN framework. It acts as a mechanism to share routes selectively and enable communication between VRFs at Layer 2 when a customer edge device is connected to a single provider edge router. By using VRF leaking, we now provide both isolation and segmentation among VRFs while still allowing inter-VRF communication through EVPN route type 2 (MAC+IP) import. This mechanism enables controlled communication between VRFs, permitting specific routes or traffic to traverse VRF boundaries while preserving the isolation of unrelated routes and traffic.

VRF leaking or stitching is a technique used in multi-VRF environments to enable communication between two or more VRFs at Layer 2. With VRF Leaking, you can interconnect VRFs at Layer 2, which allows traffic to flow between them using Layer 2 gateways, which act as bridge devices to forward traffic between VRFs.

You can connect different VRFs using a Layer 2 gateway or bridge to allow traffic between the VRFs in an EVPN single-homing mode. You can define traffic policies to allow traffic to pass between the VRFs, which includes filtering based on EVPN EVI, and/or MAC addresses. After the Layer 2 gateway and traffic policies are configured, communication between the VRFs is established at Layer 2. Layer 2 frames can now be forwarded between the VRFs while maintaining the VRF isolation for Layer 3 traffic.

Configure VRF Leaking

Perform these tasks to configure VRF route leak between global route table and VRF route table.

1. Configure BGP where the router performs the route leak.
2. Configure the route policies, these policies help you filter which prefixes are permitted to be leaked. In this example, the **route-policy GLOBAL-2-VRF** and **route-policy VRF-2-GLOBAL** are used.
3. Configure the VRF and apply the route-policy.

Configuration Example

```
/* Configure BGP */
Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 10.10.10.10
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# network 172.16.20.0/24
Router(config-bgp-af)# exit
Router(config-bgp)# address-family vpnv4 unicast
Router(config-bgp-af)# exit
Router(config-bgp)# vrf ORANGE
Router(config-bgp-vrf)# rd 100:100
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# network 192.168.10.0/24
Router(config-bgp-vrf-af)# commit

/* Configure route policies */
Router(config)# route-policy GLOBAL-2-VRF
Router(config-rpl)# if destination in (172.16.20.0/24) then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
```

```
Router(config-rpl)# end-policy
Router(config)# route-policy VRF-2-GLOBAL
Router(config-rpl)# if destination in (192.168.10.0/24 le 32) then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
Router(config-rpl)# end-policy

/* Configure VRF and apply route-policy */
Router(config)# vrf ORANGE
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import from default-vrf route-policy GLOBAL-2-VRF
Router(config-vrf-af)# import route-target
Router(config-vrf-import-rt)# 100:100
Router(config-vrf-import-rt)# exit
Router(config-vrf-af)# export to default-vrf route-policy VRF-2-GLOBAL
Router(config-vrf-af)# export route-target
Router(config-vrf-export-rt)# 100:100
Router(config-vrf-export-rt)# commit
```

Running Configuration

This section shows the VRF leaking running configuration.

```
router bgp 100
 bgp router-id 10.10.10.10
 address-family ipv4 unicast
  network 172.16.20.0/24
 !
 address-family vpnv4 unicast
 !
 vrf ORANGE
  rd 100:100
  address-family ipv4 unicast
   network 192.168.10.0/24
  !
 !
!
route-policy GLOBAL-2-VRF
  if destination in (172.16.20.0/24) then
    pass
  endif
end-policy
!
route-policy VRF-2-GLOBAL
  if destination in (192.168.10.0/24 le 32) then
    pass
  endif
end-policy
!
vrf ORANGE
 address-family ipv4 unicast
  import from default-vrf route-policy GLOBAL-2-VRF
  import route-target
   100:100
  !
  export to default-vrf route-policy VRF-2-GLOBAL
  export route-target
   100:100
  !
 !
!
```

Verification

Verify the prefixes appear in the RIB and BGP tables.

```
Router# show route
Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

C    10.88.174.0/24 is directly connected, 1d20h, MgmtEth0/RSP0/CPU0/0
L    10.88.174.223/32 is directly connected, 1d20h, MgmtEth0/RSP0/CPU0/0
L    10.10.10.10/32 is directly connected, 04:33:44, Loopback100
C    172.16.20.0/24 is directly connected, 07:03:18, HundredGigE0/0/0/24
L    172.16.20.1/32 is directly connected, 07:03:18, HundredGigE0/0/0/24
B    192.168.10.0/24 is directly connected, 03:02:21, HundredGigE0/0/0/0 (nexthop in vrf
ORANGE)

Router# show ip bgp
BGP router identifier 10.10.10.10, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 5
BGP main routing table version 5
BGP NSR Initial initsync version 3 (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
*> 172.16.20.0/24     0.0.0.0                  0         32768 i
*> 192.168.10.0/24    0.0.0.0                  0         32768 i

Processed 2 prefixes, 2 paths
```

The following show output displays the information for the VRF ORANGE:

```
Router# show route vrf ORANGE
Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

B    172.16.20.0/24 is directly connected, 01:43:49, HundredGigE0/0/0/24 (nexthop in vrf
default)
   C    192.168.10.0/24 is directly connected, 07:06:38, HundredGigE0/0/0/24
   L    192.168.10.2/32 is directly connected, 07:06:38, HundredGigE0/0/0/0
```

```
Router# show bgp vrf ORANGE
BGP VRF ORANGE, state: Active
BGP Route Distinguisher: 100:100
VRF ID: 0x60000003
BGP router identifier 10.10.10.10, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000012   RD version: 9
BGP main routing table version 9
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

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: 100:100 (default for vrf ORANGE)
*> 172.16.20.0/24     0.0.0.0                  0         32768 i
*> 192.168.10.0/24    0.0.0.0                  0         32768 i

Processed 2 prefixes, 2 paths
```

Core Isolation by Interface Tracking for EVPN Single-Homing

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
Core Isolation by Interface Tracking for EVPN Single-Homing	Release 7.11.1	You can now monitor the connectivity of the customer-facing interface on the PE router using object tracking (OT). If the interface fails or becomes unavailable, the router isolates the customer site from the rest of the EVPN network. Isolating the core from the network prevents the customer site from advertising its routes to other sites on the EVPN single-home network, ensuring that traffic isn't sent to the failed PE router. Object tracking involves continuous monitoring of network interfaces, including links or ports on routers. By actively observing the status of these interfaces, administrators can dynamically adjust the network configuration based on their availability and health. This feature is supported only on Q200-based line cards. Use Object Tracking commands to track and monitor the connected interfaces.

You can effectively isolate the core provider edge if there are any issues or failures in the connected interfaces, ensuring that the rest of the network remains unaffected and operational.

To isolate the core provider edge device, you can configure the device to track the interfaces connecting to the core provider edge. To do this, you can assign a tracking object to those interfaces. The tracking object monitors the state of the interfaces, such as link up or link down.

Object Tracking

The object that receives the action may not have any relationship with the tracked objects, and the action is based on changes in the properties of the tracked object. A specific network element, such as a static route or interface status, is considered an object tracked and monitored by the device. Each tracked object is identified by a unique name specified by the track command in configuration mode. When the state of the tracked object changes, the tracking process receives the notification. The tracked objects can have an up or down state. A

list can track multiple objects using a flexible method that combines objects with Boolean logic. This functionality includes

Object tracking (OT) is a mechanism for tracking an object to take any action on another object as configured by the user. The object that receives the action may not have any relationship with the tracked objects, and the action is based on changes in the properties of the tracked object. A specific network element, such as a static route or interface status, is considered an object tracked and monitored by the device.

Each tracked object is identified by a unique name specified by the track command in configuration mode.

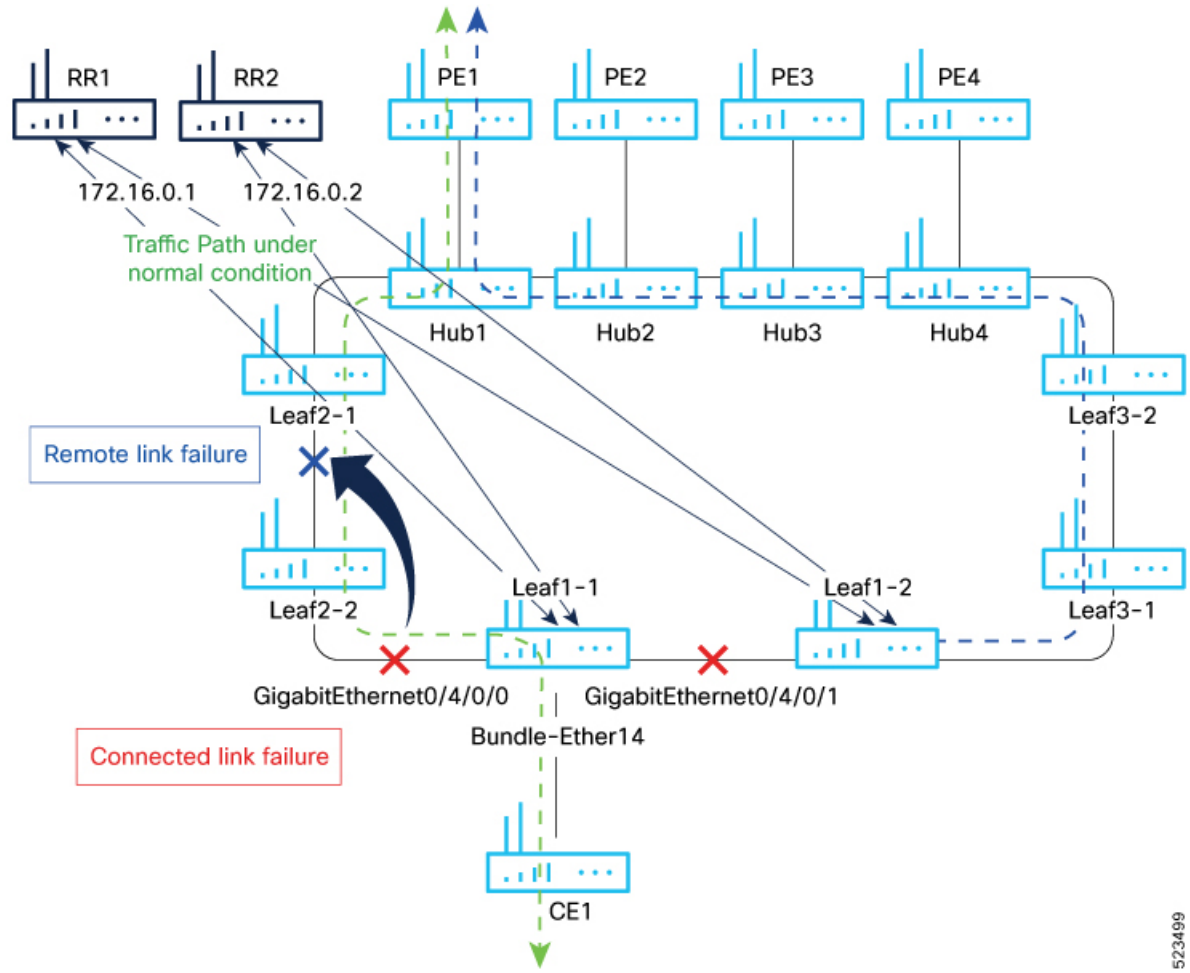
When the state of the tracked object changes, the tracking process receives the notification. The tracked objects can have an up or down state. A list can track multiple objects using a flexible method that combines objects with Boolean logic. This functionality includes:

- 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.

Here are some of the object tracking benefits:

- Provides real-time monitoring of network elements and tracks the reachability and availability of important objects in the network.
- You can automate network management tasks and implement dynamic failover or load balancing mechanisms based on the state changes of tracked objects.
- You can improve network availability by quickly detecting and responding to changes in the status of tracked objects. This enables proactive measures to be taken to minimize network downtime.
- You can define policies and actions to be triggered when the state of a tracked object changes. This allows for flexible network management and the ability to implement specific actions based on network conditions.

Figure 9: Core Isolation by Interface Tracking



Consider a traffic flow from CE1 to PE1. The CE1 sends the traffic from Leaf1-1. 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.

You can track the connected interfaces to identify the 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.

Configure EVPN Core Isolation

Perform the following tasks to configure EVPN core isolation:

1. Configure BGP
2. Track the Line Protocol State of an interface
3. Track neighbor adress-family state
4. Track objects for both interfaces and neighbors

Note

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

  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.

- 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.

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.

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 HundredGigE0/0/0/24
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 HundredGigE0/0/0/25
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 the EVPN core isolation 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 HundredGigE0/0/0/24
 !
!
track interface-2
 type line-protocol state
  interface HundredGigE0/0/0/25
 !
!
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 and track the status of interfaces and core group. The following show output examples display the status of interfaces and tracks as UP.

```
Router# show track

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 HundredGigE0/0/0/24 line-protocol
        Line protocol is UP
        2 changes, last change 20:13:32 UTC Tue May 26 2020

Track interface-2
        Interface HundredGigE0/0/0/25 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
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 HundredGigE0/0/0/24               line protocol
Up
interface-2                     interface HundredGigE0/0/0/25               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
```

The following output shows that when interfaces HundredGigE0/0/0/24 and HundredGigE0/0/0/25 go down, error-disable is triggered on the Bundle-Ether14 interface.

```
Router# show track brief
Track                           Object                                      Parameter
Value
-------------------------------------------------------------------------------------------------------
neighbor-A                      bgp nbr L2VPN EVPN 172.16.0.1 vrf defau reachability
DOWN
neighbor-B                      bgp nbr L2VPN EVPN 172.16.0.1 vrf defau reachability
```

```
DOWN
core-group-1                    list                                    boolean and
DOWN
interface-1                     interface HundredGigE0/0/0/24           line protocol
DOWN
interface-2                     interface HundredGigE0/0/0/25           line protocol
DOWN
interface-group-1               list                                    boolean or
DOWN
neighbor-group-1                list                                    boolean or
DOWN
-------------------------------------------------------------------------------------------------------

Router# show bgp track

VRF                   Address-family       Neighbor         Status     Flags
default               L2VPN EVPN           172.16.0.1       DOWN       0x01
default               L2VPN EVPN           172.16.0.2       DOWN       0x01

Processed 2 entries
-------------------------------------------------------------------------------------------------------

Router# show interfaces bundle-ether14
Bundle-Ether14 is error disabled, line protocol is administratively down
  Interface state transitions: 2
  Hardware is Aggregated Ethernet interface(s), address is 0024.f715.36c0
  Internet address is Unknown
  MTU 1514 bytes, BW 0 Kbit
     reliability 255/255, txload Unknown, rxload Unknown
  Encapsulation ARPA,
  Full-duplex, 0Kb/s
  loopback not set,
  Last link flapped 00:01:04
    No. of members in this bundle: 1
      TenGigE0/0/0/6/7             Full-duplex  10000Mb/s    Configured
  Last input 00:01:04, output 00:01:04
  Last clearing of "show interface" counters 04:44:35
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     263008898 packets input, 212215917663 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 130838604 broadcast packets, 130840312 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     787685280 packets output, 637258623124 bytes, 0 total output drops
     Output 393146795 broadcast packets, 393148545 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

EVPN Core Isolation through Peer Failure Detection

Table 14: Feature History Table

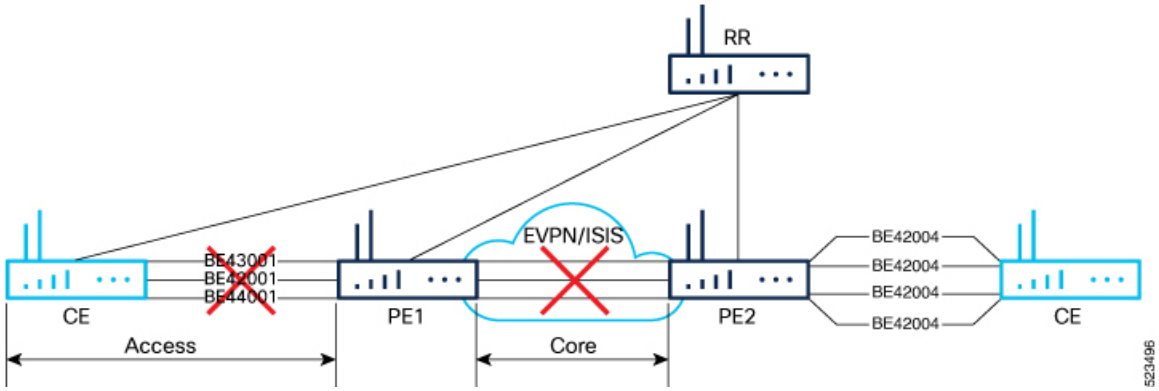
Feature Name	Release Information	Feature Description
EVPN Core Isolation through Peer Failure Detection	Release 7.11.1	You can now isolate the provider edge (PE) device from the network when there is a core link failure, preventing traffic disruptions and data leakage that could result from a compromised or malfunctioning peer. Upon detecting a link failure, the affected PE device is isolated from the core network, and the EVPN brings down the PE's Ethernet Segment (ES), which is associated with the access interface attached to the customer edge (CE) device. This feature is supported only on Q200-based line cards. The feature introduces the **core-isolation-group** command.

You can now detect a core link failure and isolate the affected PE device from the network, ensuring the continued operation and security of the EVPN network. EVPN core isolation is highly beneficial, especially in extensive deployments such as data centers. This method guarantees the stability, security, and efficiency of the EVPN network by segregating various segments or sites within the core network.

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 the access interface attached to the customer edge (CE) device.

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

Figure 10: EVPN Core Isolation Protection



When the core links of PE1 go down, the EVPN detects the link failure and isolates the PE1 node from the core network by bringing down the access network. This prevents CE from sending any traffic to PE1. Since the BGP session also goes down, the BGP invalidates all the routes that the failed PE advertised.

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

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. 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.

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.

Configuration Example

In this example, configure core interfaces under the EVPN group and associate that group to the attachment circuit (AC) connected to the EVPN single-homing CE device.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# group 42001
Router(config-evpn-group)# core interface HundredGigE 0/0/0/24
Router(config-evpn-group)# core interface HundredGigE 0/0/0/25
Router(config-evpn-group)#exit
!
Router(config-evpn)# group 43001
```

```
Router(config-evpn-group)# core interface HundredGigE 0/0/0/26
Router(config-evpn-group)# core interface HundredGigE 0/0/0/27
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 HundredGigE 0/0/0/24
   core interface HundredGigE 0/0/0/25
   !
  group 43001
   core interface HundredGigE 0/0/0/26
   core interface HundredGigE 0/0/0/27
   !
 !
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.

The following output shows that the core is isolated because the core interfaces are down, bringing down the access interfaces connected to this core.

```
Router# show evpn group
EVPN Group: 42001
  State: Isolated
  Core Interfaces:
     HundredGigE 0/0/0/24: down
     HundredGigE 0/0/0/25: down

  Access Interfaces:
     Bundle-Ether42001: down
```

The following output shows that the core is in the Ready state because both the core and access interfaces are UP.

```
Router# show evpn group
EVPN Group: 43001
  State: Ready
  Core Interfaces:
     HundredGigE 0/0/0/26: up
     HundredGigE 0/0/0/27: up

  Access Interfaces:
     Bundle-Ether43001: up
```

MAC Mobility for EVPN Single-Homing

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
MAC Mobility for EVPN Single-Homing	Release 7.11.1	You can now seamlessly move MAC addresses between various network devices or locations while preserving their connectivity and associated network services. This ensures uninterrupted communication for devices or virtual machines frequently changing their physical or virtual location within the network. The L2 gateway dynamically updates its forwarding table when a MAC address moves from one device to another within the EVPN E-LAN network, guaranteeing that packets destined for that MAC address are correctly forwarded to its new location. This feature is supported only on Q200-based line cards.

MAC Mobility provides the flexibility to move devices or virtual machines to different physical hosts or locations within the network, which enables efficient resource utilization by enabling dynamic distribution of traffic and optimized routing decisions based on the location of MAC addresses. This feature is valuable in scenarios such as data centers, where virtual machines or containers must be able to move across physical hosts without disrupting their network connectivity.

Feature Highlights

- Facilitates seamless movement of MAC addresses among different devices or network locations, maintaining uninterrupted connectivity. This agility allows devices or virtual machines to be flexible and mobile, accommodating dynamic workloads and efficient resource allocation.
- Manages a substantial volume of mobile devices or virtual machines, permitting seamless movement across different network segments without causing disruptions or requiring manual reconfiguration.

- Ensures that packets are appropriately forwarded to the updated MAC address locations, optimizing routing decisions, curbing unnecessary traffic, and enhancing overall network performance.
- Eliminates the need for manual configuration changes when devices or virtual machines move within the network. This simplifies network management and reduces the likelihood of human errors or misconfigurations.

MAC mobility supports:

Detect and Block Duplicate MAC Addresses

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
Detect and Block Duplicate MAC Addresses	Release 7.11.1	You can now effectively mitigate traffic disruptions, packet loss, and potential network outages in your network operations by detecting and freezing duplicate MAC addresses and blocking all associated routes. This feature is supported only on Q200-based line cards. The feature introduces the **host mac-address duplicate-detection** command.

The duplicate MAC address detection feature automatically checks any host with a duplicate MAC address and blocks all routes that have a duplicate MAC address, as hosts with duplicate MAC addresses can cause unnecessary churn in a network and result in traffic loss for either or both hosts sharing the same MAC address.

Multiple devices sharing identical MAC addresses can cause MAC address flapping, which intermittently results in different devices claiming the same MAC address and causing network devices to update their forwarding tables constantly. Duplicate MAC addresses may lead to excessive network traffic, increased CPU utilization on network devices, and overall instability.

Based on the predefined parameters, the router identifies and freezes a duplicate MAC address. However, you can use a configuration option to prevent the MAC address from being permanently frozen, offering flexibility in managing network configurations.

The router keeps track of MAC addresses as they move from one host to another, handling the mobility of EVPN hosts. The router learns and relearns MAC routes from both hosts if two hosts have the same MAC address. Each time it learns the MAC route from one host, it counts as one move, as the newly learned route supersedes the previous route learned from the other host. This process continues back and forth until the router marks the MAC address as a duplicate based on the configured parameters. Use the **host mac-address duplicate-detection** command to configure the router when to mark a MAC address as duplicate and whether to freeze or unfreeze it as it moves between different hosts using the following configurable parameters.

- **move-count**: The number of times a MAC address has changed its location within a specified period between different hosts to be considered a duplicate. The period is specified in the **move-interval** parameter.

- **move-interval**: The duration within which a MAC address moves a certain number of times between different hosts to be considered as duplicate and frozen temporarily. This number is specified in the **move-count** parameter.
- **freeze-time**: The length of time a MAC address is locked after it has been detected as a duplicate. After this period, the MAC address is unlocked and it is allowed to learn again.
- **retry-count**: The number of times a MAC address is unlocked after it has been detected as a duplicate before it is frozen permanently.

A syslog notifies the user that the particular MAC address is frozen. While a MAC address is frozen, any new MAC routes or updates to existing MAC routes with the frozen MAC address are ignored. After the **freeze-time** has elapsed, the corresponding MAC routes are unfrozen and the value of the **move-count** is reset to zero. For any unfrozen local MAC routes, an ARP probe and flush are initiated while the remote MAC routes are put in the probe mode. This restarts the duplicate detection process.

The router also maintains the information about the number of times a particular MAC address has been frozen and unfrozen. If a MAC address is marked as duplicate after it is unfrozen **retry-count** times, it is frozen permanently. However, you can unfreeze permanently frozen hosts using any of the following recommended procedures to clear frozen hosts:

- Shut down the host which is causing duplicate traffic.
- Use the **clear l2route evpn frozen-mac frozen-flag** command to clear the frozen hosts.

How to Prevent MAC Address Freezing

You can unfreeze the permanently frozen MAC addresses with a configurable option to enable a MAC address to undergo infinite duplicate detection and recovery cycles without being frozen permanently. The MAC address is permanently frozen when duplicate detection and recovery events occur three times within a 24-hour window. If any of the duplicate detection events happen outside the 24-hour window, the MAC address undergoes only one duplicate detection event and all previous events are ignored.

Use the **host mac-address duplicate-detection retry-count infinity** command to prevent freezing of the duplicate MAC address permanently.

The 24-hour check for consecutive duplicate detection and recovery events before permanent freezing is enabled by default. Use the **host mac-address duplicate-detection reset-freeze-count-interval** command to configure a nondefault interval after which the retry-count is reset. The range is from is one hour to 48 hours. The default is 24 hours.

Configure Duplicate MAC Address Detection and Prevent MAC Address Freezing

You can set configurable parameters to mark the host as duplicate.

- The default settings allow for five instances of duplication within 180 seconds, and the route freezes after three cycles of duplication.
- If a host moves five times within 180 seconds under the default settings, it is marked as duplicate for 30 seconds.
- During this time, route advertisements for the duplicate host will be suppressed. After 30 seconds, the host will no longer be considered a duplicate.
- If a host is detected as a duplicate for the fourth time, it will be permanently frozen, and all route advertisements will be suppressed.

Configuration Example

Perform this task to configure duplicate MAC address detection and prevent MAC address freezing.

Use the **host mac-address duplicate-detection** command and set the configurable parameters on the router to mark a MAC address as duplicate as it moves between different hosts.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# host mac-address duplicate-detection
Router(config-evpn-host-mac-addr-dup-detection)# move-count 2
Router(config-evpn-host-mac-addr-dup-detection)# freeze-time 10
Router(config-evpn-host-mac-addr-dup-detection)# retry-count 2
Router(config-evpn-host-mac-addr-dup-detection)# commit
```

Use the **retry-count infinite** command to prevent freezing of duplicate MAC address permanently.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# host MAC-address duplicate-detection
Router(config-evpn-host-mac-addr-dup-detection)# retry-count infinite
Router(config-evpn-host-mac-addr-dup-detection)# commit
```

Running Configuration

This section shows the running configuration of duplicate MAC address detection and preventing MAC address freezing.

```
evpn
 host mac-address duplicate-detection
  move-count 2
  freeze-time 10
  retry-count 2
!
evpn
 host mac-address duplicate-detection
  retry-count infinite
 !
```

Verification

In this example, the *0011.0000.0001* MAC address is identified as duplicate and subsequently frozen. The *DmZm* flag denotes that the MAC address has been marked as duplicate and frozen.

```
Router# show l2route evpn mac-ip 10.47.177.225 detail
Topo ID  Mac Address    IP Address      Producer    Next Hop(s)
   Seq No   Flags
Opaque Data Type        Opaque Data Len
Opaque Data Value
Opaque NH Type          Opaque NH Len
Opaque NH Value
-------- -------------- --------------- ----------- --------------------------------------
 -------- --------
161      0011.0000.0001 10.47.177.225   LOCAL       Bundle-Ether8.1212, N/A
   43       BLDmZm
N/A                     N/A
N/A
N/A                     N/A
N/A
    Last Update: Fri Nov 03 17:42:09.426 CET
```

```
161      0011.0000.0001 10.47.177.225   L2VPN       25000/I/ME, N/A
   42       DmZm
0                       12
0x06000000 0x3b010080 0x00000000
```

EVPN E-Tree

Table 17: Feature History Table

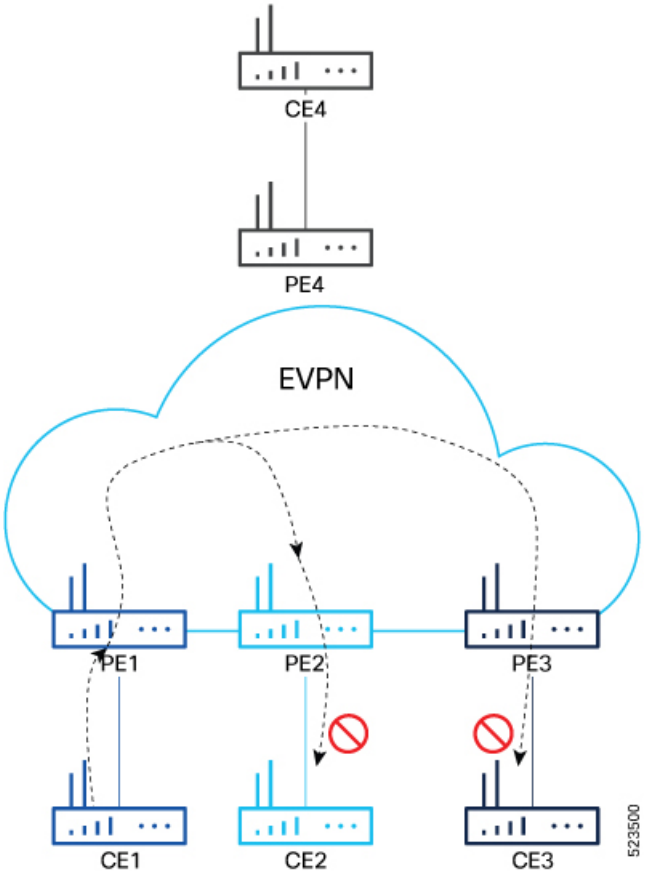
Feature Name	Release Information	Feature Description
EVPN E-Tree	Release 7.11.1	We now enable efficient forwarding of ethernet traffic in a tree-like topology where a root PE router broadcasts or multicasts traffic to all the leaf PE routers while the leaf PE routers only forward traffic destined for the respective customer sites connected to them. This feature is supported only on Q200-based line cards. The feature introduces the **etree rt-leaf** command.

Now, you have the ability to segregate traffic between the central hub and individual remote sites, significantly enhancing network security. Traffic segregation prevents direct communication between remote sites, thereby minimizing network congestion and reducing surface attacks, making it valuable for enterprises and service providers in diverse networking scenarios.

The EVPN Ethernet Tree (E-Tree) service enables you to define attachment circuits (ACs) as either root or leaf nodes. Here is how it works:

- The root node (PE4) is the central point of the E-Tree service, typically located within the service provider's network. The root node serves as the communication hub, establishing connections with all leaf nodes.
- Leaf nodes (PE1, PE2, and PE3) are the endpoints of the E-Tree service, representing customer sites or remote locations. Leaf nodes communicate with the root node and exchange data with others. However, direct communication between leaf nodes is restricted.
- E-Tree service uses EVPN, facilitating dynamic learning of MAC addresses across the network and enabling efficient and flexible communication between the root and leaf nodes.
- The root node forwards traffic to all the leaf nodes, but each leaf node does not forward traffic to other leaf nodes. This isolation ensures that communication between the root and all leaf nodes is maintained without creating unnecessary traffic between the leaf nodes.
- If a PE is not configured as an E-Tree leaf, it is considered as a root by default.

Figure 11: EVPN E-Tree



You can implement E-Tree in either of the following ways:

- Scenario 1 - All ACs at a particular PE for a given EVI or BD can be either root or leaf site, and all traffic for an EVI from a PE in the network is from either a root or a leaf. In this scenario, you have two options to configure E-Tree:
  - Scenario 1a - You can configure E-Tree with route-targets (RT) constraints using two RTs per EVI.
  - Scenario 1b - You can configure E-Tree without route-targets (RT) constraints and using the **etree leaf** label.
- Scenario 2 - You configure a PE device to have both root and leaf sites for a given EVI.

Note Only Scenario 1a is supported.

Scenario 1a

EVPN E-Tree using RT constraints, lets you configure BGP RT import and export policies for an attachment circuit. You can define communication between the leaf and root nodes. The attachment circuit (AC) of a bridge domain (BD) or the remote PE node can send L2 traffic to the provider edge (PE) nodes. L2

communication can only happen from root to leaf and leaf to root for a given BD. This feature prevents L2 communication between the ACs of two or more leafs. Every EVI uses two BGP RTs. The root ACs and leaf ACs are each associated with a separate set of RTs.

Rules for Import and Export Policies under the BGP of EVPN EVI Instances

- Root PE exports its ROOT-RT using the BGP export policy. It also imports other ROOT-RT from the corresponding root PE for the same EVI. This is necessary where there is more than one root for a particular BD and EVPN EVI, for example, in a multihome active-active scenario or multihome port-active and single-active scenarios.
- Root PE imports LEAF-RT using the BGP import policy for an EVPN EVI. This enables the root to be aware of all remote L2 MAC addresses through EVPN RT2 advertisement of leaf PE node for a given E-Tree EVI.
- Leaf PE exports its LEAF-RT using the BGP export policy to let the root be aware of the reachability of its directly connected L2 endpoints through EVPN RT2 advertisement.
- Leaf PE imports ROOT-RT using the BGP import policy. It helps the leaf to know about the L2 endpoints, which are reachable through the AC of BD under EVPN EVI instance of root PE. You must not import LEAF-RT using the BGP Import policy to avoid L2 Communication between two leaf PEs.

The BGP import and export policies applies to all EVPN RTs along with the RT2 advertisement.

MAC Address Learning

- L2 MAC addresses are learned on AC of a particular BD on leaf PE as type LOCAL. The same MAC address is advertised to root PE as EVPN RT2. On the remote root PE, the MAC table replicates the entry of the MAC address with the learning type L2VPN. Also, it associates the MPLS label of its BGP peer, which advertises RT2 to the root PE node.
- L2 MAC addresses are learned on AC of a particular BD on the root as type LOCAL. The same MAC address is advertised to peer root or leaf PE as EVPN RT2. On the remote root PE or leaf PE, the MAC table replicates the entry of the MAC address with the learning type L2VPN. Also, it associates the MPLS label of its BGP peer, which advertises RT2 to the PE node.
- L2 MAC addresses are learned on AC of a particular BD on the root as type LOCAL. The MAC table of the peer root node synchronizes the replicated entry of the MAC address with the learning type as L2VPN for the same ESI and with the same AC as the next hop. This avoids flooding and duplication of known unicast traffic.

Configure EVPN E-Tree (Scenario 1a)

Perform the following tasks on the PE device where you want to configure E-tree:

1. Configure bridge domain
2. Configure attachment circuit
3. Configure EVPN EVI
4. Configure bundle Ethernet
5. Configure EVPN interface

Configuration Example

```
/* Configure bridge domain */
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-Ether700.305
Router (config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface Bundle-Ether720.305
Router (config-l2vpn-bg-bd-ac)# exit
Router (config-l2vpn-bg-bd)# evi 305
Router (config-l2vpn-bg-bd-evi)# commit

/* Configure attachment circuit */
Router# configure
Router(config)# interface Bundle-Ether700.305 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 305
Router(config-l2vpn-subif)# rewrite ingress tag pop 1 symmetric
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether720.305 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 305
Router(config-l2vpn-subif)# rewrite ingress tag pop 1 symmetric
Router(config-l2vpn-subif)# commit

/* Configure EVPN EVI */
Router# configure
Router(config)# evpn
Router(config-evpn)# evi 305
Router(config-evpn-instance)# bgp
Router(config-evpn-instance-bgp)# route-target import 1001:305>>  Route target of leaf
Router(config-evpn-instance-bgp)# route-target export 1001:5305>> Route target of root
Router(config-evpn-instance-bgp)# exit
Router(config-evpn-instance)# exit
Router(config-evpn-instance)# etree
Router(config-evpn-instance-etree)# rt-leaf
Router(config-evpn-instance-etree)# exit
Router(config-evpn-instance)# control-word-disable
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance-mac)# commit

/* Configure bundle Ethernet */
Router# configure
Router(config)# interface Bundle-Ether700
Router(config-if)# lacp system mac 00aa.aabb.1010
Router(config-if)# lacp switchover suppress-flaps 300
Router(config-if)# lacp cisco enable link-order signaled
Router(config-if)# bundle wait-while 100
Router(config-if)# exit
Router(config)# interface Bundle-Ether720
Router(config-if)# lacp system mac 00aa.aabb.1212
Router(config-if)# lacp switchover suppress-flaps 300
Router(config-if)# lacp cisco enable link-order signaled
Router(config-if)# bundle wait-while 100
Router(config-if)# commit

/* Configure EVPN interface */
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether700
Router(config-evpn-ac)# ethernet-segment
```

```
Router(config-evpn-ac-es)# identifier type 0 00.00.00.00.00.00.00.00.00
Router(config-evpn-ac-es)# bgp route-target 0000.0000.0001
Router(config-evpn-ac-es)# exit
Router(config-evpn-ac)# exit
Router(config-evpn)# exit
Router(config-evpn)# interface Bundle-Ether720
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.00.00.00.00.00.00.00.00
Router(config-evpn-ac-es)# bgp route-target 0000.0000.0020
Router(config-evpn-ac-es)# commit
```

Running Configuration

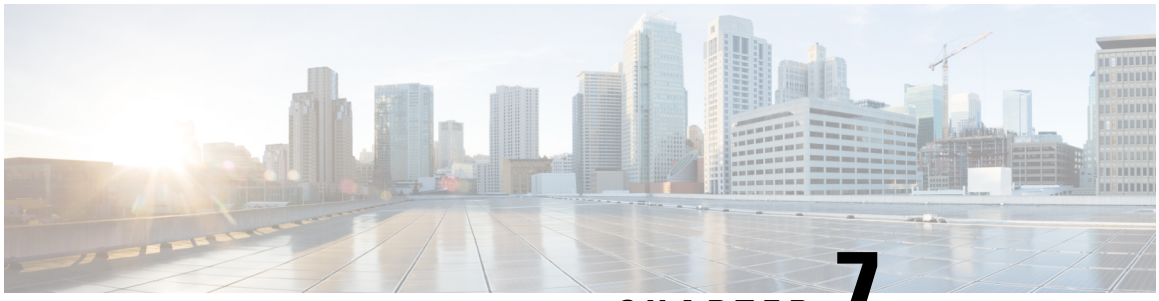
```
l2vpn
 bridge group BG1
  bridge-domain BD1
   interface Bundle-Ether700.305
   !
   interface Bundle-Ether720.305

   !
   evi 305
   !
  !
 !
interface Bundle-Ether700.305 l2transport
 encapsulation dot1q 305
 rewrite ingress tag pop 1 symmetric
!
interface Bundle-Ether720.305 l2transport
 encapsulation dot1q 305
 rewrite ingress tag pop 1 symmetric
!
evpn
 evi 305
  bgp
   route-target import 1001:305
   route-target export 1001:5305
  !
  etree
   rt-leaf
  !
  control-word-disable
  advertise-mac
  !
 !
!
interface Bundle-Ether700
 lacp system mac 00aa.aabb.1010
 lacp switchover suppress-flaps 300
 lacp cisco enable link-order signaled
 bundle wait-while 100
!
interface Bundle-Ether720
 lacp system mac 00aa.aabb.1212
 lacp switchover suppress-flaps 300
 lacp cisco enable link-order signaled
 bundle wait-while 100
!
evpn
 interface Bundle-Ether700
  ethernet-segment
   identifier type 0 00.00.00.00.00.00.00.00.00
```

```
   bgp route-target 0000.0000.0001
  !
 !
!
evpn
 interface Bundle-Ether720
  ethernet-segment
   identifier type 0 00.00.00.00.00.00.00.00.00
   bgp route-target 0000.0000.0020
  !
 !
!
```

The single-homing PE device only knows about its local L2 MAC addresses and the MAC addresses learned on the root node. The leaf single-homing PE device does not know any other MAC addresses learned on other leaf PE nodes. Each leaf is completely isolated from other leaf PEs in terms of their knowledge of MAC addresses learned from each other.

```
Router$ show l2route evpn mac all
Topo ID  Mac Address    Producer    Next Hop(s)
-------- -------------- ----------- --------------------------------------
200      0011.0100.0001 L2VPN       30579/I/ME, N/A
200      0011.0100.0002 L2VPN       30579/I/ME, N/A
200      0011.0100.0003 L2VPN       30579/I/ME, N/A
200      0011.0100.0004 L2VPN       30579/I/ME, N/A
200      0011.0100.0005 L2VPN       30579/I/ME, N/A
200      0012.0100.0001 LOCAL       Bundle-Ether700.305, N/A
200      0012.0100.0002 LOCAL       Bundle-Ether700.305, N/A
200      0012.0100.0003 LOCAL       Bundle-Ether700.305, N/A
200      0012.0100.0004 LOCAL       Bundle-Ether700.305, N/A
200      0012.0100.0005 LOCAL       Bundle-Ether700.305, N/A
```



CHAPTER 7

Seamless Migration of VPLS Network to EVPN Network

This chapter describes how to migrate VPLS network to EVPN network.

Seamless Migration of VPLS Network to EVPN Network

Table 18: Feature History Table

Feature Name	Release Information	Feature Description
Seamless Migration of VPLS Network to EVPN Network	Release 7.11.1	You can now provision EVPN service on existing VPLS-enabled PEs individually, thus ensuring a seamless VPLS-to-EVPN migration without traffic disruption. This feature is supported only on Q200-based line cards.

Although VPLS is a widely deployed Layer 2 VPN technology, customers prefer to migrate their VPLS network to EVPN to leverage the scaling benefits and ease of deployment. Recognizing the significance of preserving investments in VPLS, certain service providers seek ways to seamlessly connect their existing VPLS networks with the new networks running EVPN.

You can now migrate the PE nodes from legacy VPLS to EVPN gradually and incrementally without any service disruption.

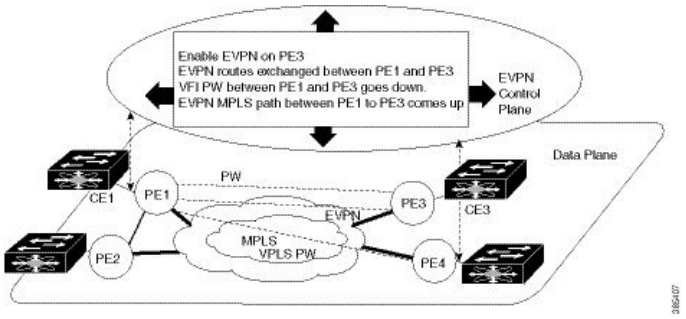
Instead of performing a network-wide software upgrade at the same time on all PEs, this feature provides the flexibility to migrate one PE at a time. Thus allows the coexistence of legacy VPLS and EVPN-VPLS dual-stack in the core for a given L2 attachment circuit (AC) over the same MPLS network.

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

Seamless migration 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 12: Seamless Migration of VPLS Network to EVPN Network



You can introduce the EVPN service to all the selected VPLS provider edge (PE) nodes simultaneously. However, to avoid traffic disruption, provision EVPN service on existing VPLS-enabled PEs one by one.

- To migrate from VPLS to EVPN, enable EVPN in a VPN instance of VPLS service on PE1, which starts advertising the EVPN inclusive multicast route to other PE nodes.

  Since no inclusive multicast routes are received from other PE nodes, VPLS pseudowires between PE1 and other PE nodes remain active.

- PE1 forwards traffic using VPLS pseudowires and advertises all MAC addresses learned from CE1 using EVPN route type-2.
- Next, enable EVPN on PE3, and it starts advertising an inclusive multicast route to other PE nodes.
- PE1 and PE3 discover each other through EVPN routes and shut down pseudowires between them.

  EVPN service replaces VPLS service between PE1 and PE3.

- PE1 keeps running VPLS service with PE2 and PE4 and starts EVPN service with PE3 in the same VPN instance called EVPN seamless integration with VPLS.
- Migrate the remaining PE nodes until all four PE nodes are enabled with the EVPN service.
- Eventually, the VPLS service is completely replaced with the EVPN service in the network, and all VPLS pseudowires are shut down.

Configure EVPN on the Existing VPLS Network

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

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

Configure L2 EVPN Address-Family

Configure BGP on the PE routers and enable EVPN address family under both BGP and participating neighbors.

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
   !
 !
```

Verification

Verify if the BGP neighbor is functional.

```
Router# show bgp l2vpn evpn summary
BGP router identifier 200.0.1.1, 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 1
BGP NSR Initial initsync version 4294967295 (Not Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.
```

```
Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               1          1          1          0           1           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
200.0.4.1         0     65530       2       2        0    0    0 00:00:09          0
```

Configure EVI under EVPN Configuration Mode

To enable EVPN on PE1, configure EVI. 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)#advertise-mac
Router(config-evpn-evi)#commit
```

Running Configuration

```
configure
 evpn
  evi
   advertise-mac
   !
  !
 !
```

Verification

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

```
Router#show evpn summary
----------------------------
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 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
```

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 HundredGigE0/0/0/0
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 172.16.0.1 pw-id 12
Router(config-l2vpn-bg-bd-vfi-pw)#neighbor 192.168.0.1 pw-id 13
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 HundredGigE0/0/0/0
    !
    evi 1
    !
   vfi v1
    neighbor 172.16.0.1 pw-id 12
    neighbor 192.168.0.1 pw-id 13
     mpls static label local 20001 remote 10001
     !
    !
```

Verification

Verify the EVPN and VPLS status.

```
Router# show l2vpn bridge-domain
Legend: pp = Partially Programmed.
Bridge group: vplstoevpn, bridge-domain: vplstoevpn, 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: 2 (1 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
List of EVPNs:
  EVPN, state: up
List of ACs:
  Hu0/0/0/0, state: up, Static MAC addresses: 0, MSTi: 5
List of Access PWs:
List of VFIs:
  VFI vpls (up)
    Neighbor 172.16.0.1 pw-id 12, state: down, Static MAC addresses: 0
    Neighbor 192.168.0.1 pw-id 13, state: up, Static MAC addresses: 0
```

The output indicates that the VPLS PW "neighbor 172.16.0.1 pw-id 12" is replaced by EVPN service, as the EVPN control plane discovered that both local PE and remote PE (172.16.0.1) have enabled EVPN service on the L2VPN instance.