EVPN Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 24.1.x, 24.2.x, 24.3.x

First Published: 2024-03-14

Last Modified: 2024-09-04

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

© 2024 Cisco Systems, Inc. All rights reserved.

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

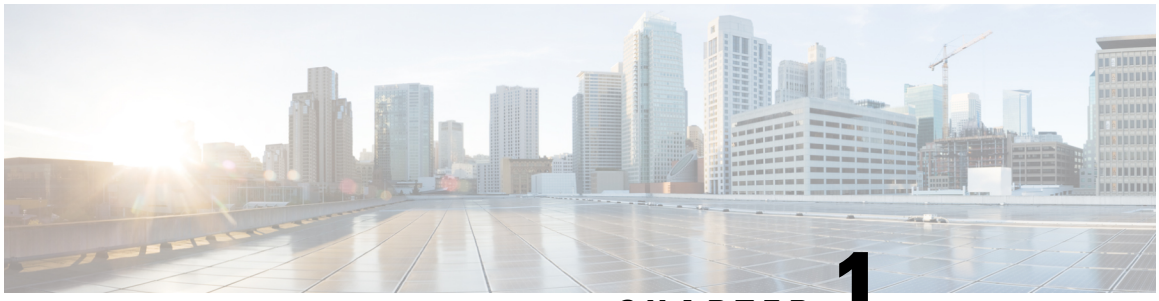
Date	Summary
September 2024	Republished with feature updates for Release 24.3.1.
June 2024	Republished with feature updates for Release 24.2.11.
March 2024	Initial release of this document.

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 24.x.x

Feature	Description	Changed in Release	Where Documented
Layer 2 Fast Reroute	This feature was introduced.	Release 24.3.1	Layer 2 Fast Reroute, on page 142
BUM Ingress Replication for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	BUM Ingress Replication for EVPN E-LAN, on page 105
CFM on EVPN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	CFM on EVPN, on page 147
Core Isolation by Interface Tracking for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	Core Isolation by Interface Tracking for EVPN E-LAN, on page 114

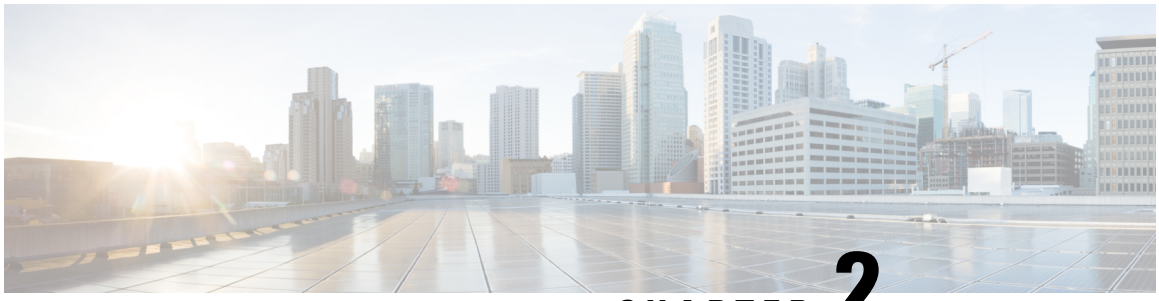
Feature	Description	Changed in Release	Where Documented
Detect and Block Duplicate MAC Addresses on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	Detect and Block Duplicate MAC Addresses, on page 126
Ethernet VPN Virtual Private Wire Service on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN Virtual Private Wire Service (VPWS) - Ethernet Line (E-Line) Service, on page 26
EVPN Core Isolation through Peer Failure Detection on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN Core Isolation through Peer Failure Detection, on page 122
EVPN Cost-Outon 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN Cost-Out, on page 101
EVPN Designated Forwarder Election on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN Designated Forwarder Election, on page 84
EVPN E-LAN L2 Gateway Single-Homing on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E.	This feature was modified.	Release 24.3.1	EVPN E-LAN L2 Gateway Single-Homing, on page 17
EVPN E-LAN Single-Flow-Active Multi-Homing on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN E-LAN Single-Flow-Active Multi-Homing, on page 72

Feature	Description	Changed in Release	Where Documented
EVPN E-Tree (Scenario 2) on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN E-Tree for EVPN E-LAN, on page 130
EVPN MPLS Multi-Homing on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN MPLS Multi-Homing, on page 55
EVPN Multiple Services per Ethernet Segment on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN Multiple Services per Ethernet Segment, on page 137
EVPN Seamless Integration with Legacy VPWS on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	EVPN Seamless Integration with Legacy VPWS, on page 29
MAC Mobility for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	MAC Mobility for EVPN E-LAN, on page 125
Seamless Migration of VPLS Network to EVPN Network on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	Seamless Migration of VPLS Network to EVPN Network, on page 20
Split-Horizon Groups for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	Split-Horizon Groups for EVPN E-LAN, on page 107

Feature	Description	Changed in Release	Where Documented
Virtual Ethernet Segment on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	Virtual Ethernet Segment, on page 96
VRF Leaking for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	This feature was modified.	Release 24.3.1	VRF Leaking for EVPN E-LAN, on page 110
BUM Ingress Replication for EVPN E-LAN on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	BUM Ingress Replication for EVPN E-LAN, on page 105
CFM on EVPN	This feature was introduced.	Release 24.2.11	CFM on EVPN, on page 147
Core Isolation by Interface Tracking for EVPN E-LAN on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	Core Isolation by Interface Tracking for EVPN E-LAN, on page 114
Detect and Block Duplicate MAC Addresses on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	Detect and Block Duplicate MAC Addresses, on page 126
Ethernet VPN Virtual Private Wire Service on the Q200 and 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	EVPN Virtual Private Wire Service (VPWS) - Ethernet Line (E-Line) Service, on page 26
EVPN Core Isolation through Peer Failure Detection on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	EVPN Core Isolation through Peer Failure Detection, on page 122
EVPN Cost-Out	This feature was introduced.	Release 24.2.11	EVPN Cost-Out, on page 101
EVPN Designated Forwarder Election	This feature was introduced.	Release 24.2.11	EVPN Designated Forwarder Election, on page 84
EVPN E-LAN L2 Gateway Single-Homing on the 88-LC1-36EH Line Cards.	This feature was modified.	Release 24.2.11	EVPN E-LAN L2 Gateway Single-Homing, on page 17

Feature	Description	Changed in Release	Where Documented
EVPN E-LAN Single-Flow-Active Multi-Homing	This feature was introduced.	Release 24.2.11	EVPN E-LAN Single-Flow-Active Multi-Homing, on page 72
EVPN E-Tree (Scenario 2)	This feature was modified.	Release 24.2.11	EVPN E-Tree for EVPN E-LAN, on page 130
EVPN MPLS Multi-Homing	This feature was introduced.	Release 24.2.11	EVPN MPLS Multi-Homing, on page 55
EVPN Multiple Services per Ethernet Segment	This feature was introduced.	Release 24.2.11	EVPN Multiple Services per Ethernet Segment, on page 137
EVPN Seamless Integration with Legacy VPWS on the Q200 and 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	EVPN Seamless Integration with Legacy VPWS, on page 29
MAC Mobility for EVPN E-LAN on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	MAC Mobility for EVPN E-LAN, on page 125
Seamless Migration of VPLS Network to EVPN Network on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	Seamless Migration of VPLS Network to EVPN Network, on page 20
Split-Horizon Groups for EVPN E-LAN on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	Split-Horizon Groups for EVPN E-LAN, on page 107
Virtual Ethernet Segment	This feature was introduced.	Release 24.2.11	Virtual Ethernet Segment, on page 96
VRF Leaking for EVPN E-LAN on the 88-LC1-36EH Line Cards	This feature was modified.	Release 24.2.11	VRF Leaking for EVPN E-LAN, on page 110

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.

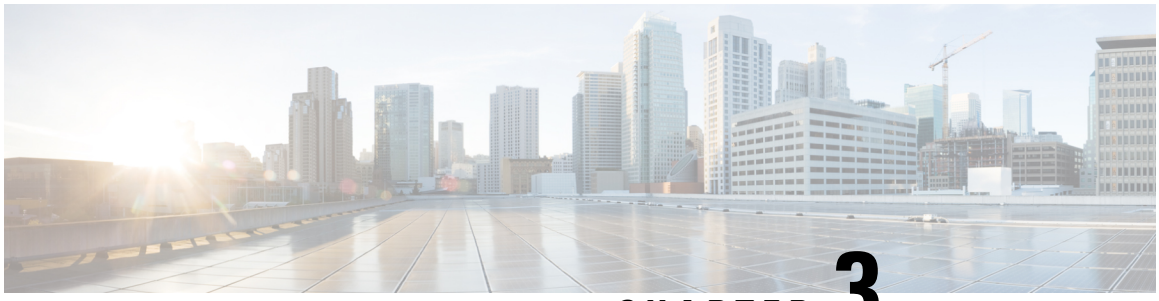
Using YANG Data Models

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 MPLS

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.

Line cards and routers with the Q100, Q200, and P100 based Silicon One ASIC support this feature.

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.

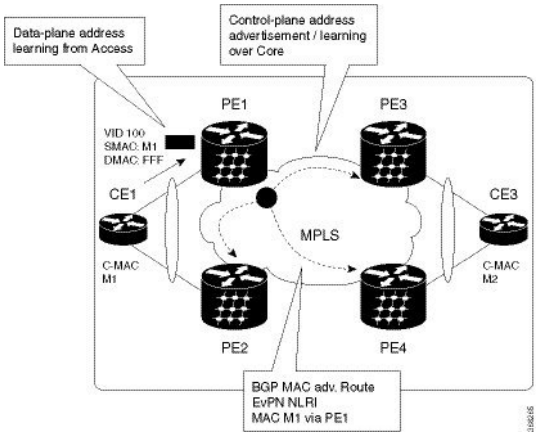
EVPN Operation

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.

Note The unknown unicast suppression (**unknown-unicast-suppression**) operation under a given EVI is not supported on the Cisco 8000 platform.

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.
- Multi-homing - Enables you to connect a customer edge (CE) device to more than one provider edge (PE) device. Multi-homing ensures redundant connectivity. The redundant PE device ensures that there is no traffic disruption when there is a network failure.

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

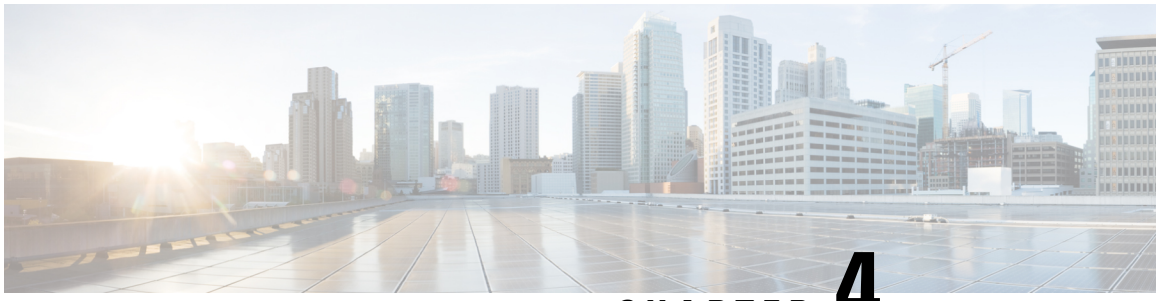
Note

- 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 MPLS Single Homing

This chapter describes how to configure EVPN MPLS 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 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	EVPN single-homing is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN E-LAN L2 Gateway Single-Homing on the 88-LC1-36EH Line Cards	Release 24.2.11	EVPN single-homing is now supported on routers with the 88-LC1-36EH line cards.

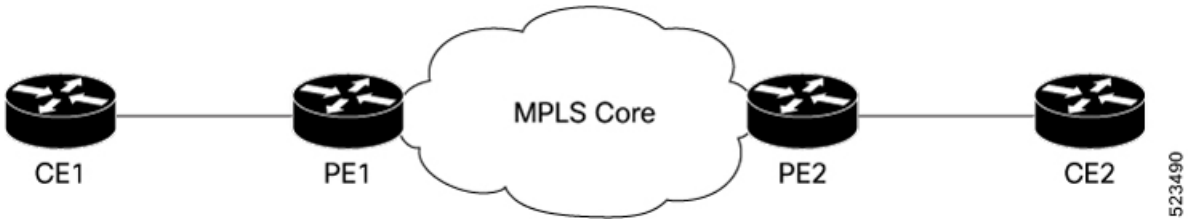
Feature Name	Release Information	Feature Description
EVPN E-LAN L2 Gateway Single-Homing	Release 7.11.1	We now offer a cost-effective and simplified solution for seamless communication between various customer sites connected to the same service provider network using Ethernet Virtual Private Network (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. Disable EVPN multi-homing onbundle interface.
2. Set up BGP for L2VPN and EVPN
3. Configure bridge domain
4. 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


/* By default, the bundle interface is in EVPN multi-homing mode.
To disable EVPN multi-homing, configure bundle-Ether AC with ESI value (identifier type)
set to zero. */
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether1
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 multi-homing globally */

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

/* Configure MAC advertisement. */
Router(config)# evpn
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
    ethernet-segment
    identifier type 0 00.00.00.00.00.00.00.00.00
!
!
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
```

Seamless Migration of VPLS Network to EVPN Network

Table 6: Feature History Table

Feature Name	Release Information	Feature Description

Seamless Migration of VPLS Network to EVPN Network on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The seamless VPLS-to-EVPN migration is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Seamless Migration of VPLS Network to EVPN Network on the 88-LC1-36EH Line Cards	Release 24.2.11	The seamless VPLS-to-EVPN migration is now supported on routers with the 88-LC1-36EH line cards.
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.

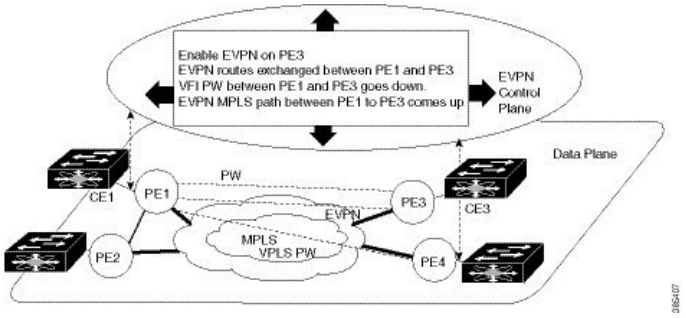
You can now migrate the PE nodes from legacy VPLS to EVPN gradually and incrementally without any service disruption.

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 2: 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.

EVPN Virtual Private Wire Service (VPWS) - Ethernet Line (E-Line) Service

The terms Virtual Private Wire Service (VPWS) and Ethernet Line (E-Line) Service are used interchangeably.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet VPN Virtual Private Wire Service on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The EVPN VPWS or E-Line service is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Ethernet VPN Virtual Private Wire Service on the Q200 and 88-LC1-36EH Line Cards	Release 24.2.11	The EVPN VPWS or E-Line service is now supported on routers with the Q200 and 88-LC1-36EH line cards.
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 (E-Line) technology works on IP and MPLS core; IP core to support BGP and MPLS core for switching packets between the endpoints.

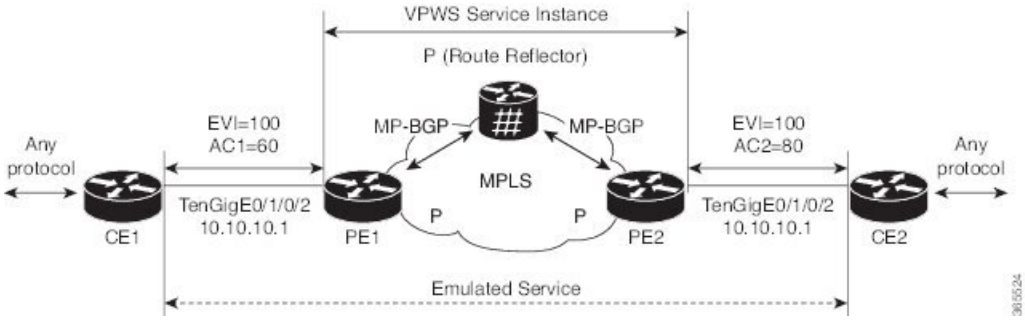
EVPN-VPWS Single Homed

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 3: 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.

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 8: Feature History Table

Feature Name	Release Information	Feature Description
EVPN Seamless Integration with Legacy VPWS on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The seamless migration of VPWS to EVPN-VPWS services on PE nodes is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN Seamless Integration with Legacy VPWS on the Q200 and 88-LC1-36EH Line Cards	Release 24.2.11	The seamless migration of VPWS to EVPN-VPWS services on PE nodes is now supported on routers with the Q200 and 88-LC1-36EH line cards.
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.

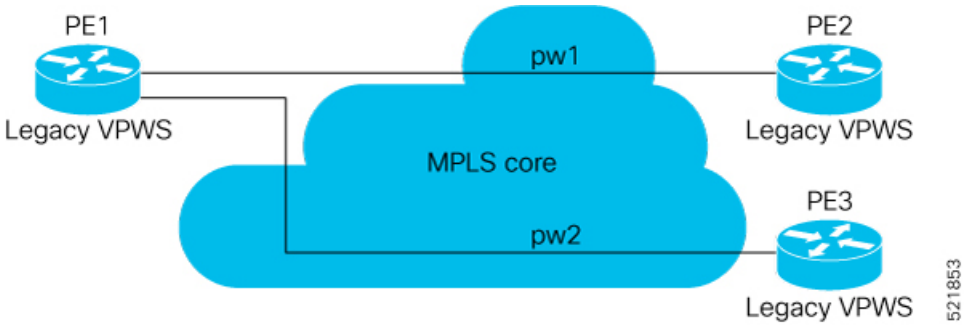
In an EVPN-VPWS network, VPN instances are grouped by EVPN Instance VPN ID (EVI) and identified by an ethernet tag or attachment circuit ID (AC-ID). EVI is also associated with route-targets and route-distinguisher.

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 4: VPWS Nodes

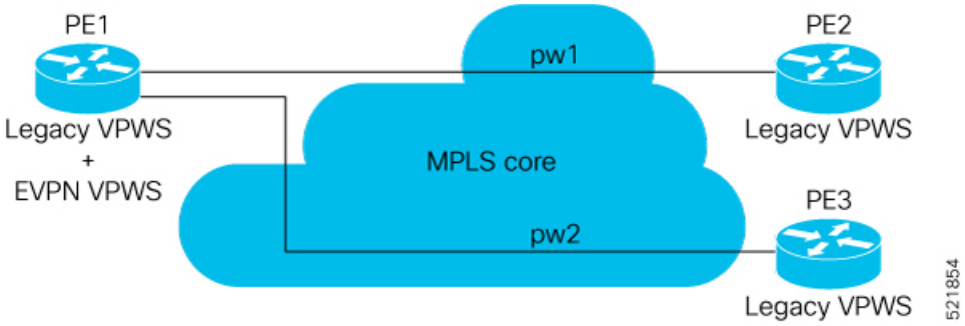


In this topology, PE1, PE2, PE3 are provider edge devices in the MPLS network and the legacy VPWS cross-connects are up and running between PE1, PE2, and PE3.

- 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 5: PE1 Enabled with EVPN-VPWS

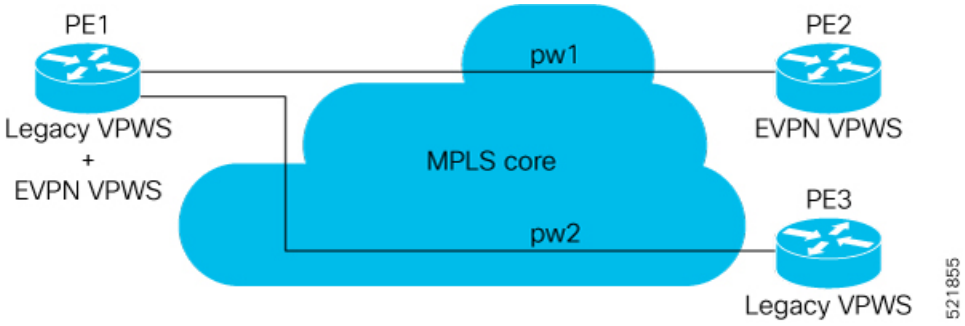


Let's understand what happens when only PE1 is migrated to EVPN-VPWS:

- 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 6: 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 tasks:

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 tasks:

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 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 9: 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.

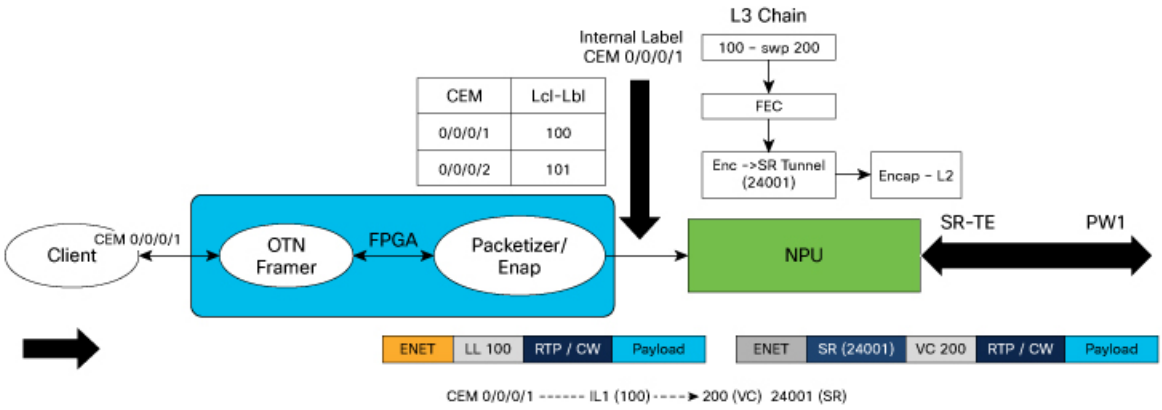
CEM helps PLE endpoints to provide native client interfaces. CEM service is a method through which data can be transmitted over Ethernet or MPLS networks. CEM over a packet carries circuits over Packet Switched Network (PSN) placing the client bitstreams into packet payload with appropriate pseudowire 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 7: PLE Forwarding Flow – Imposition



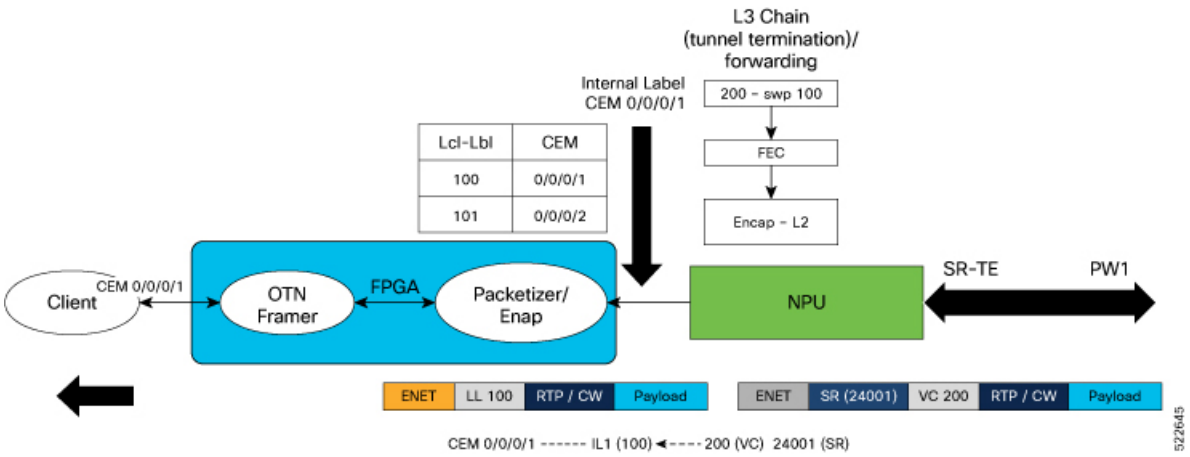
In the diagram, traffic from client may be of any port mode like FC, OTN, SDH, SONET, or Ethernet. Field Programmable Gate Array (FPGA) acts as a forwarding block. FPGA sends the traffic from the client towards NPU with an assigned internal local label.

- 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 8: 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*.

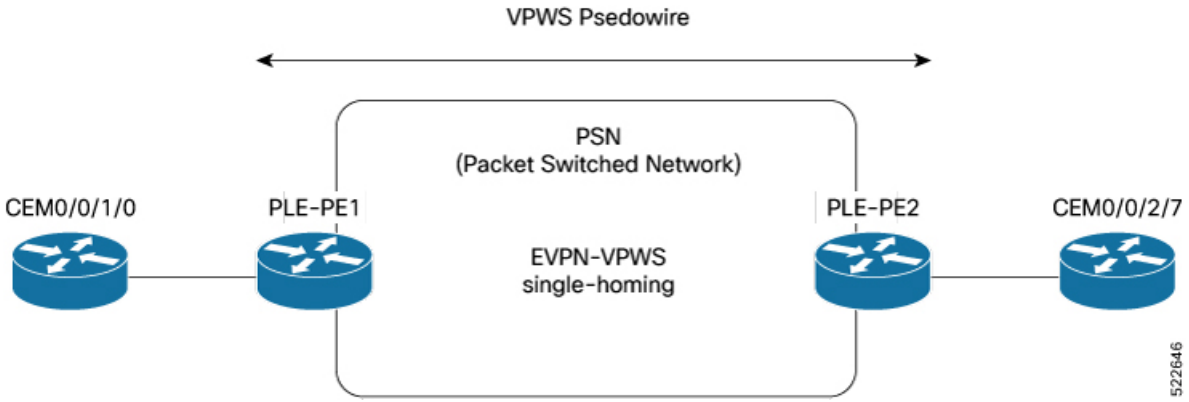
Configure PLE over EVPN VPWS

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 9: 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:

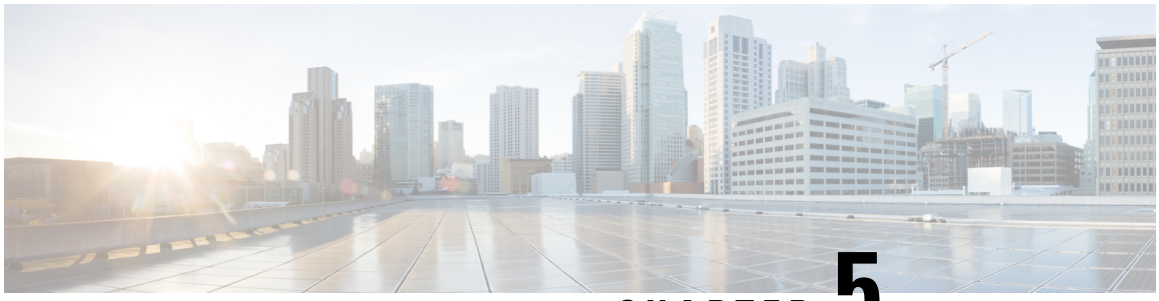
- Cisco-IOS-XR-controller-odu-cfg.yang
- Cisco-IOS-XR-controller-otu-cfg.yang
- Cisco-IOS-XR-controller-sonet-cfg.yang
- Cisco-IOS-XR-drivers-icpe-ethernet-cfg.yang

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 5

EVPN MPLS Multi-Homing

This chapter describes how to configure EVPN MPLS Multi-Homing on EVPN E-LAN and EVPN E-Line services.

EVPN MPLS Multi-Homing

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
EVPN MPLS Multi-Homing on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The EVPN MPLS multi-homing is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN MPLS Multi-Homing	Release 24.2.11	EVPN multi-homing enables you to connect a customer edge (CE) device to two or more provider edge (PE) devices to provide redundant connectivity. When the primary link fails, the standby PE device becomes active immediately, ensuring no traffic disruption and providing faster convergence. This feature is supported only on routers with the 88-LC1-36EH line cards.

In EVPN MPLS multi-homing mode, you can connect a customer edge (CE) device to more than one provider edge (PE) device to ensure redundant connectivity. The redundant PE device ensures that there is no traffic disruption when there is a network failure.

The following types of multi-homing are supported:

- Single-Active - In this mode, only a single PE among a group of PEs attached to the particular Ethernet-Segment is allowed to forward traffic to and from that Ethernet Segment.
- All-Active - In this mode, all PEs attached to the particular Ethernet-Segment is allowed to forward traffic to and from that Ethernet Segment.
- Port-Active - In this mode, only the PE which is in the active mode sends and receives the traffic. This mode supports single-active redundancy load balancing at the port-level or the interface-level.
- Single-Flow-Active - In this mode, only the PE that first advertises the host MAC address in a VLAN forwards the traffic in a specific flow.

EVPN E-LAN Single-Active Multi-Homing

In EVPN E-LAN single-active mode, the PE nodes locally connected to an Ethernet Segment load balance traffic to and from the Ethernet Segment based on EVPN service instance (EVI). Within an EVPN service instance, only one PE forwards traffic to and from the Ethernet Segment.

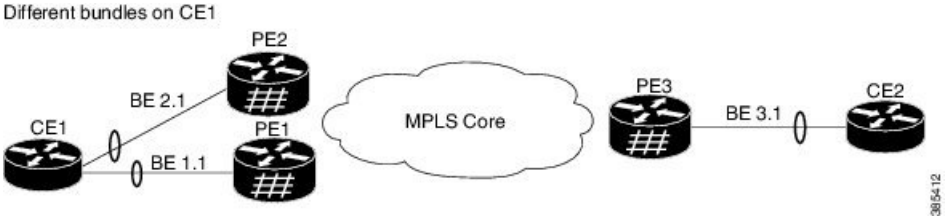
In a ring topology, only one of the PEs, which is the active PE, sends and receives the traffic to prevent a traffic loop. When the link to the active PE fails, the traffic switches over to the standby PE. Traffic switchover takes a while because the standby PE has to learn the MAC addresses of the connected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

EVPN E-LAN single-active multi-homing enables you to connect a customer edge (CE) device to more than one provider edge (PE) device with redundant connectivity. In single-active mode, only a single PE among a group of PEs attached to the particular Ethernet segment is allowed to forward traffic to and from that Ethernet segment. In this mode, only the PE that first advertises the host MAC address in a VLAN forwards the traffic in a specific flow.

When the primary link fails, the traffic quickly switches to the standby PE that learns the MAC address from the originated path, thereby providing fast convergence. This also enables load balancing of traffic to and from the Ethernet Segment based on EVPN service instance (EVI).

The following image illustrates an EVPN E-LAN single-active multi-homing topology.

Figure 10: EVPN E-LAN Single-Active Multi-Homing



In this topology, CE1 is multi-homed to PE1 and PE2. The PE1 and PE2 are connected to PE3 through MPLS core. CE2 is connected to PE3 through an Ethernet interface bundle. PE1 and PE2 advertise Type 4 routes to elect the designated forwarder (DF). The non-DF blocks the traffic in both the directions in single-active mode. In this example, PE1 is elected as the DF and PE2 is the non-DF.

The traffic flow from CE1 and CE2 happens as follows:

- CE1 sends an address resolution protocol (ARP) broadcast request to both PE1 and PE2.
- As PE1 is the designated forwarder for the EVI, PE1 forwards the ARP request from CE1.
- PE2, the non-DF, drops the traffic from CE1.
- All the traffic is sent through PE1. PE2 remains as a standby device and traffic is not sent through PE2.
- PE1 advertises MAC routes to PE3 and PE3 always sends and receives traffic through PE1.
- PE3 sends the traffic to CE2 over Ethernet interface bundle.

When there is a link failure and the active PE1 goes down, PE2 becomes active to continue with the traffic flow.

Configure EVPN Single-Active Multi-homing

Perform the following configuration on PE1, PE2, and PE3.

1. Configure BGP session and MPLS Label Distribution Protocol (LDP) to enable EVPN.
2. Configure EVPN EVI parameters and advertisement of MAC routes.
3. Enter the bundle interface mode and configure the Ethernet segment identifier (ESI) for the interface.
4. Ensure that you configure the same ESI on all the PEs.
5. Enable single-active mode by using the **load-balancing-mode single-active** command.

Configuration Example

```
/* PE1 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 54.54.54.54
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 51.51.51.51
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 55.55.55.55
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 54.54.54.54
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* PE2 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 55.55.55.55
Router(config-bgp)# address-family l2vpn evpn
```

```
Router(config-bgp)# neighbor 51.51.51.51
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 54.54.54.54
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 55.55.55.55
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* PE3 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 51.51.51.51
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 54.54.54.54
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 55.55.55.55
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 51.51.51.51
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* Configure Bridge Domain and EVI on PE1, PE2, and PE3 */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether1.1
Router(config-l2vpn-bg-bd-ac)# evi 1
Router(config-l2vpn-bg-bd-ac)# root

Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg2
Router(config-l2vpn-bg)# bridge-domain bd2
Router(config-l2vpn-bg-bd)# interface Bundle-Ether1.2
Router(config-l2vpn-bg-bd-ac)# evi 2

/* Configure EVPN EVI and advertise the MAC routes on PE1, PE2, and PE3 */
Router(config)# evpn
Router(config-evpn)# evi 1
Router(config-evpn-evi)# advertise-mac
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 2
Router(config-evpn-evi)# advertise-mac


/* Configure the same ESI on all the PE Routers */
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether1
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 40.00.00.00.00.00.00.00.01
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# commit
```

Running Configuration

```
/* PE1 Configuration */
router bgp 100
 bgp router-id 54.54.54.54
 address-family l2vpn evpn
 !
 neighbor 51.51.51.51
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 55.55.55.55
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 54.54.54.54
 interface FourHundredGigE0/0/0/2
 !
!

/* PE2 Configuration */
router bgp 100
 bgp router-id 55.55.55.55
 address-family l2vpn evpn
 !
 neighbor 51.51.51.51
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 54.54.54.54
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 55.55.55.55
 interface FourHundredGigE0/0/0/2
 !
!

/* PE3 Configuration */
router bgp 100
 bgp router-id 51.51.51.51
 address-family l2vpn evpn
 !
 neighbor 54.54.54.54
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 55.55.55.55
```

```
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 51.51.51.51
 interface FourHundredGigE0/0/0/3
 !
!

/* Configuration on all the PEs */
l2vpn
 bridge group bg1
  bridge-domain bd1
   interface Bundle-Ether1.1
   !
   evi 1
   !
  !
 !
 bridge group bg2
  bridge-domain bd2
   interface Bundle-Ether1.2
   !
   evi 2
   !
  !
 !
!
evpn
 evi 1
  advertise-mac
  !
 !
 evi 2
  advertise-mac
  !
 !
 interface Bundle-Ether1
  ethernet-segment
   identifier type 0 40.00.00.00.00.00.00.00.01
   load-balancing-mode single-active
  !
 !
!
```

Verification

The following output shows configuration of single-active mode on PE1:

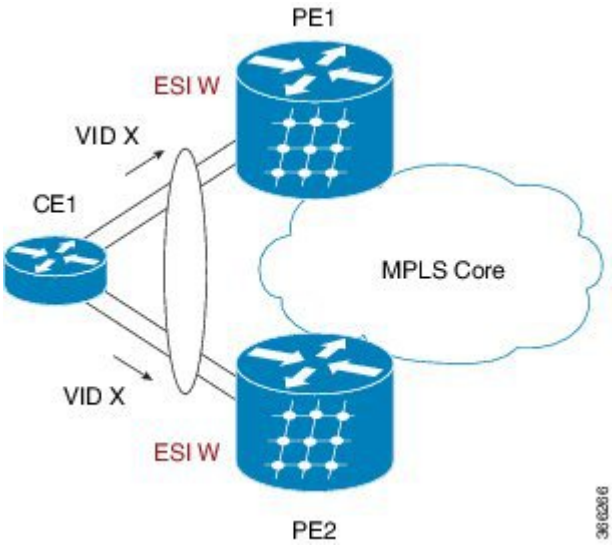
```
Router#show evpn ethernet-segment interface BE1 carving detail


Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0040.0000.0000.0000.0001 BE1                               54.54.54.54
                                                           55.55.55.55
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether1
```

```
   Interface MAC  : 008d.9c38.7205
   IfHandle       : 0x0f00003c
   State          : Up
   Redundancy     : Not Defined
ESI ID            : 1
ESI type          : 0
   Value          : 0040.0000.0000.0000.0001
ES Import RT      : 4000.0000.0000 (from ESI)
Topology          :
   Operational    : MH, Single-active
   Configured     : Single-active (AApS)
Service Carving   : Auto-selection
   Multicast      : Disabled
Convergence       :
Peering Details   : 2 Nexthops
   54.54.54.54 [MOD:P:00:T]
   55.55.55.55 [MOD:P:00:T]
Service Carving Synchronization:
   Mode           : NONE
   Peer Updates   :
           54.54.54.54 [SCT: N/A]
           55.55.55.55 [SCT: 2024-03-12 10:42:30.1710254]
Service Carving Results:
   Forwarders     : 2
   Elected        : 1
          EVI E   :        2
   Not Elected    : 1
          EVI NE  :        1
EVPN-VPWS Service Carving Results:
   Primary        : 0
   Backup         : 0
   Non-DF         : 0
MAC Flush msg     : STP-TCN
Peering timer     : 3 sec [not running]
Recovery timer    : 30 sec [not running]
Carving timer     : 0 sec [not running]
Revert timer      : 0 sec [not running]
HRW Reset timer   : 5 sec [not running]
Local SHG label   : 24004
Remote SHG labels : 1
            24004 : nexthop 55.55.55.55
Access signal mode: Bundle OOS
```

EVPN E-LAN All-Active Multi-Homing

In all-active multi-homing mode, a device is connected to multiple PEs and all the links actively forward the traffic on that Ethernet Segment.

Figure 11: EVPN E-LAN All-Active Multi-homing



All-active load-balancing is known as Active/Active per Flow (AApF). In the above topology, identical Ethernet Segment Identifier is used on both EVPN PEs. PEs are attached to Ethernet Segment using bundle interfaces. In the CE, single bundles are configured towards two EVPN PEs. In this mode, both PE1 and PE2 can forward the traffic within the same EVI.

Configure EVPN E-LAN All-Active Multi-Homing

Perform the following tasks on both PE1 and PE2.

1. Configure BGP session and MPLS Label Distribution Protocol (LDP) to enable EVPN.
2. Configure EVPN EVI parameters and advertisement of MAC routes.
3. Enter the bundle interface mode and configure the Ethernet segment identifier (ESI) for the interface.
4. Ensure that you configure the same ESI on all the PEs.

Configuration Example

```
/* PE1 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 54.54.54.54
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 51.51.51.51
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 55.55.55.55
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
```

```
/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 54.54.54.54
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* PE2 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 55.55.55.55
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 51.51.51.51
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 54.54.54.54
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 55.55.55.55
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* Configure Bridge Domain and EVI on PE1 and PE2 */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether11.1
Router(config-l2vpn-bg-bd-ac)# evi 1
Router(config-l2vpn-bg-bd-ac)# root

Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg2
Router(config-l2vpn-bg)# bridge-domain bd2
Router(config-l2vpn-bg-bd)# interface Bundle-Ether11.2
Router(config-l2vpn-bg-bd-ac)# evi 2

/* Configure EVPN EVI and advertise the MAC routes on PE1 and PE2 */
Router(config)# evpn
Router(config-evpn)# evi 1
Router(config-evpn-evi)# advertise-mac
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 2
Router(config-evpn-evi)# advertise-mac


/* Configure the same ESI on all the PE Routers */
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether11
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 40.00.00.00.00.00.00.00.01
Router(config-evpn-ac-es)# commit
```

Running Configuration

```
/* PE1 Configuration */
router bgp 100
 bgp router-id 54.54.54.54
 address-family l2vpn evpn
 !
 neighbor 51.51.51.51
```

```
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 55.55.55.55
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 54.54.54.54
 interface FourHundredGigE0/0/0/2
 !
!


/* PE2 Configuration */
router bgp 100
 bgp router-id 55.55.55.55
 address-family l2vpn evpn
 !
 neighbor 51.51.51.51
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 54.54.54.54
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 55.55.55.55
 interface FourHundredGigE0/0/0/2
 !
!

/* Configuration on all the PEs */
l2vpn
 bridge group bg1
  bridge-domain bd1
   interface Bundle-Ether11.1
   !
   evi 1
   !
  !
 !
 bridge group bg2
  bridge-domain bd2
   interface Bundle-Ether11.2
   !
   evi 2
   !
  !
 !
!
evpn
 evi 1
  advertise-mac
```

```
  !
 !
 evi 2
  advertise-mac
  !
 !
 interface Bundle-Ether11
  ethernet-segment
   identifier type 0 40.00.00.00.00.00.00.00.01
  !
 !
!
```

Verification

The following output shows configuration of all-active mode:

```
Router#show evpn ethernet-segment int Bundle-Ether 11 carving detail

Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0040.0000.0000.0000.0001 BE11                               54.54.54.54
                                                            55.55.55.55
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether11
     Interface MAC  : 008d.9c38.7205
     IfHandle       : 0x0f00001c
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0040.0000.0000.0000.0001
  ES Import RT      : 4000.0000.0000 (from ESI)
  Topology          :
     Operational    : MH, All-active
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 2 Nexthops
     54.54.54.54 [MOD:P:00:T]
     55.55.55.55 [MOD:P:00:T]
  Service Carving Synchronization:
     Mode           : NONE
     Peer Updates   :
             54.54.54.54 [SCT: N/A]
             55.55.55.55 [SCT: N/A]
  Service Carving Results:
     Forwarders     : 2
     Elected        : 1
            EVI E   :        2
     Not Elected    : 1
            EVI NE  :        1
  EVPN-VPWS Service Carving Results:
     Primary        : 0
     Backup         : 0
     Non-DF         : 0
  MAC Flush msg     : STP-TCN
  Peering timer     : 3 sec [not running]
  Recovery timer    : 30 sec [not running]
  Carving timer     : 0 sec [not running]
```

```
Revert timer      : 0 sec [not running]
HRW Reset timer   : 5 sec [not running]
Local SHG label   : 24004
Remote SHG labels : 1
            24004 : nexthop 55.55.55.55
Access signal mode: Bundle OOS
```

EVPN E-LAN Port-Active Multi-Homing

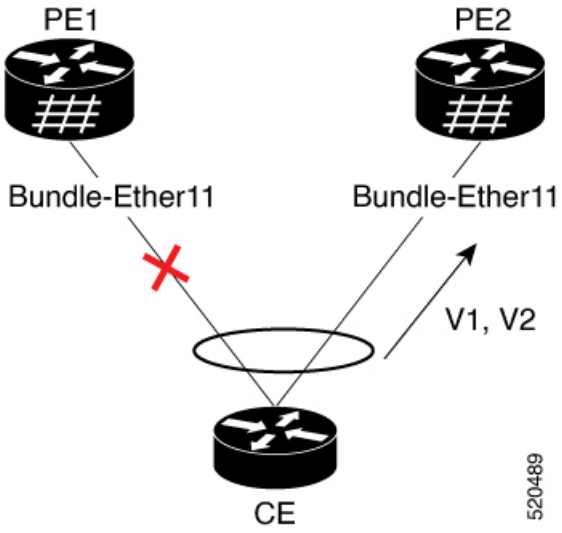
The EVPN E-LAN port-active multi-homing feature supports single-active redundancy load balancing at the port-level or the interface-level. You can use this feature when you want to forward the traffic to a specific interface, rather than have a per-flow load balancing across multiple PE routers. This feature provides a faster convergence during a link failure. This feature enables protocol simplification as only one of the physical ports is active at a given time. You can enable this feature only on bundle interfaces.

EVPN E-LAN port-active provides protocol simplification compared to Inter-Chassis Communication Protocol (ICCP), which runs on top of Label Distribution Protocol (LDP). You can use this feature as an alternative to multi-chassis link aggregation group (MC-LAG) with ICCP.

Also, you can use this feature when you want certain QoS features to work.

This feature allows one of the PEs to be in active mode and another in the standby mode at the port-level. Only the PE which is in the active mode sends and receives the traffic. The other PE remains in the standby mode. The PEs use the Designated Forwarder (DF) election mechanism to determine which PE must be in the active mode and which must be in the standby mode. You can use either modulo or Highest Random Weight (HRW) algorithm for per port DF election. By default, the modulo algorithm is used for per port DF election.

Figure 12: EVPN E-LAN Port-Active Multi-Homing



Consider a topology where the customer edge device (CE) is multi-homed to provider edge devices, PE1 and PE2. Use single link aggregation at the CE. Only one of the two interfaces is in the forwarding state, and the other interface is in the standby state. In this topology, PE2 is in the active mode and PE1 is in the standby mode. Hence, PE2 carries traffic from the CE. All services on the PE2 interface operate in the active mode. All services on the PE1 operate in the standby mode.

If you remove the port-active configuration on both PE1 and PE2 and then add back the port-active configuration on both the PEs, PE2 is chosen as an active interface again.

Configure EVPN Port-Active Multi-homing

Perform the following configuration on PE1 and PE2.

1. Configure BGP session and MPLS Label Distribution Protocol (LDP) to enable EVPN.
2. Configure EVPN EVI parameters and advertisement of MAC routes.
3. Enter the bundle interface mode and configure the Ethernet segment identifier (ESI) for the interface.
4. Ensure that you configure the same ESI on all the PEs.
5. Enable single-active mode by using the **load-balancing-mode port-active** command.

Configuration Example

```
/* PE1 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 55.55.55.55
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 51.51.51.51
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 54.54.54.54
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 55.55.55.55
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* PE2 Configuration */

/* Configure BGP Session */

Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 54.54.54.54
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 51.51.51.51
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp)# neighbor 55.55.55.55
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 54.54.54.54
Router(config-ldp)# interface FourHundredGigE0/0/0/2
```

```
/* Configure Bridge Domain and EVI on PE1 and PE2 */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether11.1
Router(config-l2vpn-bg-bd-ac)# evi 1
Router(config-l2vpn-bg-bd-ac)# root

Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg2
Router(config-l2vpn-bg)# bridge-domain bd2
Router(config-l2vpn-bg-bd)# interface Bundle-Ether11.2
Router(config-l2vpn-bg-bd-ac)# evi 2

/* Configure EVPN EVI and advertise the MAC routes on PE1 and PE2 */
Router(config)# evpn
Router(config-evpn)# evi 1
Router(config-evpn-evi)# advertise-mac
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 2
Router(config-evpn-evi)# advertise-mac


/* Configure the same ESI on all the PE Routers */
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether11
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 40.00.00.00.00.00.00.00.01
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# commit
```

Running Configuration

```
/* PE1 Configuration */
router bgp 100
 bgp router-id 55.55.55.55
 address-family l2vpn evpn
 !
 neighbor 51.51.51.51
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 54.54.54.54
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 55.55.55.55
 interface FourHundredGigE0/0/0/2
 !
!


/* PE2 Configuration */
router bgp 100
 bgp router-id 54.54.54.54
 address-family l2vpn evpn
 !
```

```
 neighbor 51.51.51.51
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
 neighbor 55.55.55.55
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
mpls ldp
 router-id 54.54.54.54
 interface FourHundredGigE0/0/0/2
 !
!

/* Configuration on all the PEs */
l2vpn
 bridge group bg1
  bridge-domain bd1
   interface Bundle-Ether11.1
   !
   evi 1
   !
  !
 !
 bridge group bg2
  bridge-domain bd2
   interface Bundle-Ether11.2
   !
   evi 2
   !
  !
 !
!
evpn
 evi 1
  advertise-mac
  !
 !
 evi 2
  advertise-mac
  !
 !
 interface Bundle-Ether11
  ethernet-segment
   identifier type 0 40.00.00.00.00.00.00.00.01
   load-balancing-mode port-active
  !
 !
!
```

Verification

The following output shows configuration of port-active mode:

```
/* PE2 is in active mode and is Up */
Router# show bundle BE11
Bundle-Ether11
  Status:                                  Up
  Local links <active/standby/configured>:   1 / 0 / 1
```

```
  Local bandwidth <effective/available>:     400000000 (400000000) kbps
  MAC address (source):                      008d.9c38.7205 (Chassis pool)
  Inter-chassis link:                        No
  Minimum active links / bandwidth:          1 / 1 kbps
  Maximum active links:                      64
  Wait while timer:                          2000 ms
  Load balancing:
    Link order signaling:                    Not configured
    Hash type:                               Default
    Locality threshold:                      None
  LACP:                                      Operational
    Flap suppression timer:                  Off
    Cisco extensions:                        Disabled
    Non-revertive:                           Disabled
  mLACP:                                     Not configured
  IPv4 BFD:                                  Not configured
  IPv6 BFD:                                  Not configured

  Port                  Device           State        Port ID         B/W, kbps
  --------------------  ---------------  -----------  --------------  ----------
  FH0/0/0/3             Local            Active       0x8000, 0x0001   400000000
      Link is Active

/* PE1 is in standby mode */

Router#show bundle BE11
Bundle-Ether11
  Status:                                    EVPN Hot-Standby
  Local links <active/standby/configured>:   0 / 1 / 1
  Local bandwidth <effective/available>:     0 (0) kbps
  MAC address (source):                      003f.ee3b.5a05 (Chassis pool)
  Inter-chassis link:                        No
  Minimum active links / bandwidth:          1 / 1 kbps
  Maximum active links:                      64
  Wait while timer:                          2000 ms
  Load balancing:
    Link order signaling:                    Not configured
    Hash type:                               Default
    Locality threshold:                      None
  LACP:                                      Operational
    Flap suppression timer:                  Off
    Cisco extensions:                        Disabled
    Non-revertive:                           Disabled
  mLACP:                                     Not configured
  IPv4 BFD:                                  Not configured
  IPv6 BFD:                                  Not configured

  Port                  Device           State        Port ID         B/W, kbps
  --------------------  ---------------  -----------  --------------  ----------
  FH0/0/0/6             Local            Standby      0x8000, 0x0001   400000000
      Link is in standby due to bundle out of service state



/* The following output shows port-active mode configuration */

Router#show evpn ethernet-segment int Bundle-Ether 11 carving detail

Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0040.0000.0000.0000.0001 BE11                              54.54.54.54
                                                           55.55.55.55
  ES to BGP Gates   : Ready
```

```
ES to L2FIB Gates : Ready
Main port         :
   Interface name : Bundle-Ether11
   Interface MAC  : 008d.9c38.7205
   IfHandle       : 0x0f00005c
   State          : Up
   Redundancy     : Not Defined
ESI ID            : 1
ESI type          : 0
   Value          : 0040.0000.0000.0000.0001
ES Import RT      : 4000.0000.0000 (from ESI)
Topology          :
   Operational    : MH
   Configured     : Port-Active
Service Carving   : Auto-selection
   Multicast      : Disabled
Convergence       :
Peering Details   : 2 Nexthops
   54.54.54.54 [MOD:P:00:T]
   55.55.55.55 [MOD:P:00:T]
Service Carving Synchronization:
   Mode           : NTP_SCT
   Peer Updates   :
           54.54.54.54 [SCT: 2024-03-12 10:58:28.1710255]
           55.55.55.55 [SCT: 2024-03-12 10:58:47.1710255]
Service Carving Results:
   Forwarders     : 2
   Elected        : 2
          EVI E   :        1,       2
   Not Elected    : 0
EVPN-VPWS Service Carving Results:
   Primary        : 0
   Backup         : 0
   Non-DF         : 0
MAC Flush msg     : STP-TCN
Peering timer     : 3 sec [not running]
Recovery timer    : 30 sec [not running]
Carving timer     : 0 sec [not running]
Revert timer      : 0 sec [not running]
HRW Reset timer   : 5 sec [not running]
Local SHG label   : 24004
Remote SHG labels : 1
            24004 : nexthop 55.55.55.55
Access signal mode: Bundle Hot-Standby
```

EVPN E-LAN Single-Flow-Active Multi-Homing

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
EVPN E-LAN Single-Flow-Active Multi-Homing on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	EVPN E-LAN single-flow-active multi-homing is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN E-LAN Single-Flow-Active Multi-Homing	Release 24.2.11	This feature introduces EVPN E-LAN single-flow-active multi-homing load balancing mode to connect PE devices in an access network that run Layer 2 access gateway protocols. In this mode, only the PE that first advertises the host MAC address in a VLAN forwards the traffic in a specific flow. When the primary link fails, the traffic quickly switches to the standby PE that learns the MAC address from the originated path, thereby providing fast convergence. The feature introduces the **load-balancing-mode** command with keyword, **single-flow-active**. This feature is supported only on routers with the 88-LC1-36EH line cards.

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

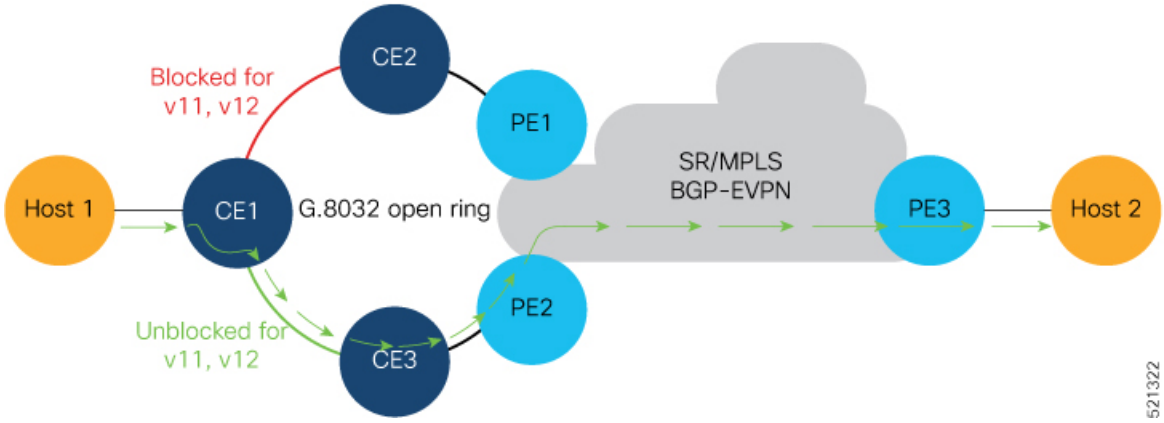
The EVPN E-LAN single-flow-active multi-homing mode connects PE devices in an access network, and in the event of active link failure the switchover happens immediately and reduces the traffic loss.

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

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

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

- In this topology, the access network devices are connected through a ring topology. The access network uses Layer-2 gateway protocols such as G.8032, MPLS-TP, REP-AG or MSTP-AG to prevent traffic loop due to continuous flooding.



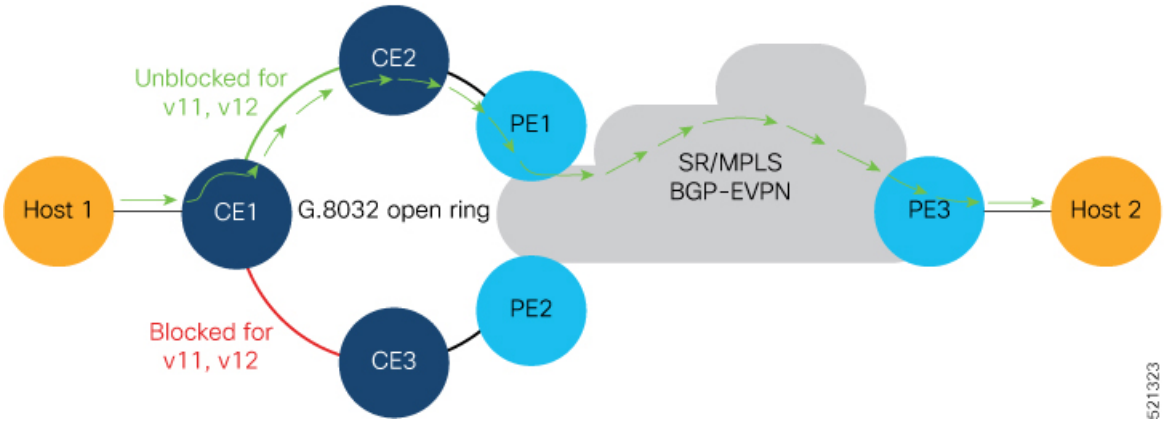
- Host 1 is connected to CE1.
- CE1 is connected to both PE1 and PE2, thus is multihomed.
- PE1 and PE2 are Multihoming devices.
- Both PE1 and PE2 is configured with the same non-zero Ethernet Segment ID (ESI) number `0 36.37.00.00.00.00.00.11.00` for the bundle interface to enable multihoming of the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

Traffic Flow

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

Failure Scenario

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



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

Limitations and Restrictions for EVPN E-LAN Single-Flow-Active Multi-Homing

- The EVPN E-LAN single-flow active multi-homing is not supported for EVPN VPWS.
- The EVPN E-LAN single-flow-active multi-homing is not supported on the Q100 and Q200 based systems.
- Starting from Release 24.1.1, only the G.8032 is supported for EVPN E-LAN single-flow-active multi-homing.

Configure EVPN E-LAN Single-Flow-Active Multi-Homing

Perform the following tasks on both PE1 and PE2.

1. Configure both PE1 and PE2 with the same EVI of 100.
2. Configure both PE1 and PE2 with the same ESI 0 36.37.00.00.00.00.00.11.01.
3. Verify the following ethernet segment status:
   a. Ensure that you configure the same ESI on both PE1 and PE2.
   b. Verify that the Single-flow-active mode is enabled in the Topology section.

Configuration Example

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

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

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

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

/* Configure bridge domain and associating the evi to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group 100
Router(config-l2vpn-bg)# bridge-domain 100
Router(config-l2vpn-bg-bd)# interface Bundle-Ether1.2
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)# evi 100
Router(config-l2vpn-bg-bd-evi)# root
Router(config)# interface Bundle-Ether1.2 l2transport
Router(config-l2vpn-subif)#encapsulation dot1q 2
Router(config-l2vpn-subif)#commit
```

Running Configuration

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

```
!
!
```

Verification

Verify the ethernet segment status:

```
Router#show  evpn  ethernet-segment  interface  be 1 detail
Legend:
B   - No Forwarders EVPN-enabled,
C   - MAC missing (Backbone S-MAC PBB-EVPN / Grouping ES-MAC vES),
RT  - ES-Import Route Target missing,
E   - ESI missing,
H   - Interface handle missing,
I   - Name (Interface or Virtual Access) missing,
M   - Interface in Down state,
O   - BGP End of Download missing,
P   - Interface already Access Protected,
Pf  - Interface forced single-homed,
R   - BGP RID not received,
S   - Interface in redundancy standby state,
X   - ESI-extracted MAC Conflict
SHG - No local split-horizon-group label allocated
Hp  - Interface blocked on peering complete during HA event
Rc  - Recovery timer running during peering sequence

Ethernet Segment Id           Interface                          Nexthops
0 36.37.00.00.00.00.00.11.01  BE1                                172.16.0.4
                                                                 172.16.0.5
ES to BGP Gates   : Ready
ES to L2FIB Gates : P
Main port         :
Interface name    : Bundle-Ether1
Interface MAC     : b0a6.51e5.00dd
IfHandle          : 0x2000802c
State             : Up
Redundancy        : Not Defined
ESI type          : 0
Value             : 07.0807.0807.0807.0800
ES Import RT      : 0708.0708.0708 (from ESI)
Source MAC        : 0000.0000.0000 (N/A)
Topology          :
Operational     : MH, Single-flow-active
Configured      : Single-flow-active
Service Carving   : Auto-selection
Multicast         : Disabled
Convergence       : MAC-Mobility
Mobility-Flush    : Debounce 1 sec, Count 0, Skip 0
                  : Last n/a
Peering Details   : 2 Nexthops
172.16.0.4 [MOD:P:00:T]
172.16.0.5 [MOD:P:00:T]
Service Carving Synchronization:
Mode              : NONE
Peer Updates      :
172.16.0.4 [SCT: N/A]
172.16.0.5 [SCT: N/A]
Service Carving Results:
Forwarders    : 1
Elected         : 0
Not Elected   : 0
EVPN-VPWS Service Carving Results:
Primary          : 0
Backup           : 0
```

```
Non-DF           : 0
MAC Flushing mode: STP-TCN
Peering timer    : 3 sec [not running]
Recovery timer   : 30 sec [not running]
Carving timer    : 0 sec [not running]
HRW Reset timer  : 5 sec [not running]
Local SHG label  : 24007
Remote SHG labels: 1
24010            : nexthop 172.16.0.5
Access signal mode: Bundle OOS (Default)
```

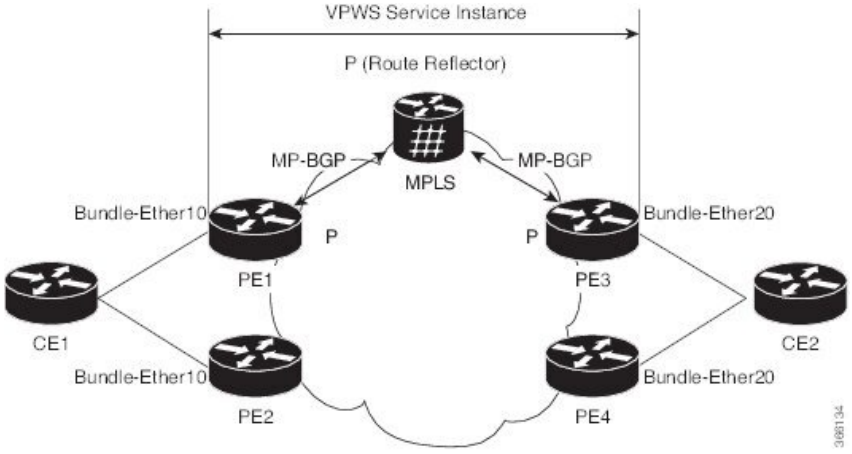
```
Router#show l2vpn protection main-interface
Main Interface ID              # of subIntf Protected  Protect Type
Bundle-Ether1                  2            Yes        ERP

Instance : 1
State    : FORWARDING

Sub-Intf # : 2

Flush    # : 6
```

EVPN E-Line Multi-Homed

The EVPN E-Line feature supports multi-homing capability that enables you to connect a customer edge device to two or more provider edge (PE) devices to provide load balancing and redundant connectivity. The load balancing is done using equal-cost multipath (ECMP).

Topology

Figure 13: EVPN E-Line Multi-Homed



In this topology, the CEs and PEs are connected as follows:

- CE1 is multi-homed to PE1 and PE2.
- CE2 is multi-homed to PE3 and PE4.
- PE1 and PE2 advertise an EAD per EVI route per AC to remote PEs which is PE3 and PE4, with the associated MPLS label. The ES-EAD route is advertised per ES (main interface), and it will not have a label.

- PE3 and PE4 advertise an EAD per EVI route per AC to remote PEs, which is PE1 and PE2, with the associated MPLS label.

Consider a traffic flow from CE1 to CE2:

- The selection of path is dependent on the CE implementation for forwarding over a LAG.
- Traffic is encapsulated at each PE and forwarded to the remote PEs (PE 3 and PE4) through MPLS core.
- Selection of the destination PE is established by flow-based load balancing.
- PE3 and PE4 send the traffic to CE2. The selection of path from PE3 or PE4 to CE2 is established by flow-based load balancing.

If there is a failure and when the link from CE1 to PE1 goes down, the PE1 withdraws the ES-EAD route; sends a signal to the remote PEs to switch all the E-Line service instances associated with this multi-homed ES to the backup PE, which is PE2.

Configure EVPN E-Line Single-Active Multi-Homed

This section describes how to configure single-active multi-homed EVPN E-Line.

- Configure cross-connect group.
- Configure point-to-point (p2p) cross-connect and assign an interface to the cross-connect.
- Enable EVPN E-Line endpoint on the p2p cross-connect.
- Configure Ethernet segment identifier (ESI) for the interface.
- Enable the single-active mode by using the **load-balancing-mode single-active** command.

```
/* PE1 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg1
Router(config-l2vpn-xc)# p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether10.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 5 source 6
Router(config-l2vpn-xc-p2p)#  root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.0a.00
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# commit

/* PE2 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg1
Router(config-l2vpn-xc)# p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether10.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 5 source 6
Router(config-l2vpn-xc-p2p)#  root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)#  ethernet-segment
```

```
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.0a.00
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# commit


/* PE3 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)#  xconnect group xg1
Router(config-l2vpn-xc)#  p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether20.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 6 source 5
Router(config-l2vpn-xc)# root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether20
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.14.00
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# commit


/* PE4 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)#  xconnect group xg1
Router(config-l2vpn-xc)#  p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether20.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 6 source 5
Router(config-l2vpn-xc)# root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether20
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.14.00
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# commit
```

Running Configuration

```
/* On PE1 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether10.2
  neighbor evpn evi 1 target 5 source 6
!
evpn
interface Bundle-Ether10
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.0a.00
   load-balancing-mode single-active
!


/* On PE2 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether10.2
  neighbor evpn evi 1 target 5 source 6
!
evpn
interface Bundle-Ether10
  ethernet-segment
```

```
   identifier type 0 00.01.00.ac.ce.55.00.0a.00
   load-balancing-mode single-active
!

/* On PE3 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether20.1
  neighbor evpn evi 1 target 6 source 5
!
evpn
interface Bundle-Ether20
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.14.00
   load-balancing-mode single-active
!

/* On PE4 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether20.1
  neighbor evpn evi 1 target 6 source 5
!
evpn
interface Bundle-Ether20
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.14.00
   load-balancing-mode single-active
!
```

Configure EVPN E-Line All-Active Multi-Homed

This section describes how to configure all-active multi-homed EVPN E-Line.

- Configure cross-connect group.
- Configure point-to-point (p2p) cross-connect and assign an interface to the cross-connect.
- Enable EVPN E-Line endpoint on the p2p cross-connect.
- Configure Ethernet segment identifier (ESI) for the interface.

```
/* PE1 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg1
Router(config-l2vpn-xc)# p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether10.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 5 source 6
Router(config-l2vpn-xc-p2p)#  root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.0a.00
Router(config-evpn-ac-es)# commit

/* PE2 Configuration */
Router# configure
```

```
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg1
Router(config-l2vpn-xc)# p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether10.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 5 source 6
Router(config-l2vpn-xc-p2p)#  root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.0a.00
Router(config-evpn-ac-es)# commit


/* PE3 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)#  xconnect group xg1
Router(config-l2vpn-xc)#  p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether20.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 6 source 5
Router(config-l2vpn-xc)# root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether20
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.14.00
Router(config-evpn-ac-es)# commit


/* PE4 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)#  xconnect group xg1
Router(config-l2vpn-xc)#  p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether20.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 6 source 5
Router(config-l2vpn-xc)# root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether20
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.14.00
Router(config-evpn-ac-es)# commit
```

Running Configuration

```
/* On PE1 */
!
configure
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether10.2
  neighbor evpn evi 1 target 5 source 6
!
evpn
interface Bundle-Ether10
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.0a.00

!


/* On PE2 */
!
configure
```

```
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether10.2
  neighbor evpn evi 1 target 5 source 6
!
evpn
interface Bundle-Ether10
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.0a.00

!

/* On PE3 */
!
configure
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether20.1
  neighbor evpn evi 1 target 6 source 5
!
evpn
interface Bundle-Ether20
  ethernet-segment
      identifier type 0 00.01.00.ac.ce.55.00.14.00

!

/* On PE4 */
!
configure
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether20.1
  neighbor evpn evi 1 target 6 source 5
!
evpn
interface Bundle-Ether20
  ethernet-segment
      identifier type 0 00.01.00.ac.ce.55.00.14.00

!
```

Configure EVPN E-Line Port-Active Multi-Homed

This section describes how to configure port-active multi-homed EVPN E-Line.

- Configure cross-connect group.
- Configure point-to-point (p2p) cross-connect and assign an interface to the cross-connect.
- Enable EVPN E-Line endpoint on the p2p cross-connect.
- Configure Ethernet segment identifier (ESI) for the interface.
- Enable the port-active mode by using the **load-balancing-mode port-active** command.

```
/* PE1 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg1
Router(config-l2vpn-xc)# p2p e1_5-6
```

```
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether10.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 5 source 6
Router(config-l2vpn-xc-p2p)#  root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.0a.00
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# commit

/* PE2 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg1
Router(config-l2vpn-xc)# p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether10.2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 5 source 6
Router(config-l2vpn-xc-p2p)#  root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.0a.00
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# commit

/* PE3 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)#  xconnect group xg1
Router(config-l2vpn-xc)#  p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether20.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 6 source 5
Router(config-l2vpn-xc)# root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether20
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.14.00
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# commit

/* PE4 Configuration */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)#  xconnect group xg1
Router(config-l2vpn-xc)#  p2p e1_5-6
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether20.1
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 6 source 5
Router(config-l2vpn-xc)# root
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether20
Router(config-evpn-ac)#  ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.01.00.ac.ce.55.00.14.00
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# commit
```

Running Configuration

```
/* On PE1 */
!
l2vpn xconnect group xg1
```

```
 p2p e1_5-6
  interface Bundle-Ether10.2
  neighbor evpn evi 1 target 5 source 6
!
evpn
interface Bundle-Ether10
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.0a.00
   load-balancing-mode port-active
!


/* On PE2 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether10.2
  neighbor evpn evi 1 target 5 source 6
!
evpn
interface Bundle-Ether10
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.0a.00
   load-balancing-mode port-active
!

/* On PE3 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether20.1
  neighbor evpn evi 1 target 6 source 5
!
evpn
interface Bundle-Ether20
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.14.00
   load-balancing-mode port-active
!

/* On PE4 */
!
l2vpn xconnect group xg1
 p2p e1_5-6
  interface Bundle-Ether20.1
  neighbor evpn evi 1 target 6 source 5
!
evpn
interface Bundle-Ether20
  ethernet-segment
   identifier type 0 00.01.00.ac.ce.55.00.14.00
   load-balancing-mode port-active
!
```

EVPN Designated Forwarder Election

Table 12: Feature History Table

Feature Name	Release Information	Feature Description

EVPN Designated Forwarder Election on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The EVPN Designated Forwarder election is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN Designated Forwarder Election	Release 24.2.11	Designated Forwarder (DF) election enables the access network to control EVPN PE devices by defining the backup path much before the event of a link failure. During the link failure, the PE node is aware of the next PE that will take over the active role and this reduces the traffic loss. DF election supports preference-based and access-driven mechanism. This feature is supported only on routers with the 88-LC1-36EH line cards.

Designated Forwarder (DF) election enables the access network to control EVPN PE devices by defining the backup path much before the event of a link failure. During the link failure, the PE node is aware of the next PE that will take over the active role and this reduces the traffic loss.

You can configure the DF election as preference-based or access driven.

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

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

Consider an interface in an access network that connects PE nodes with the EVPN PE in the core network. When this interface fails, there may be a traffic loss for a longer duration. The delay in convergence is because the backup PE is not chosen before failure occurs.

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

Restrictions

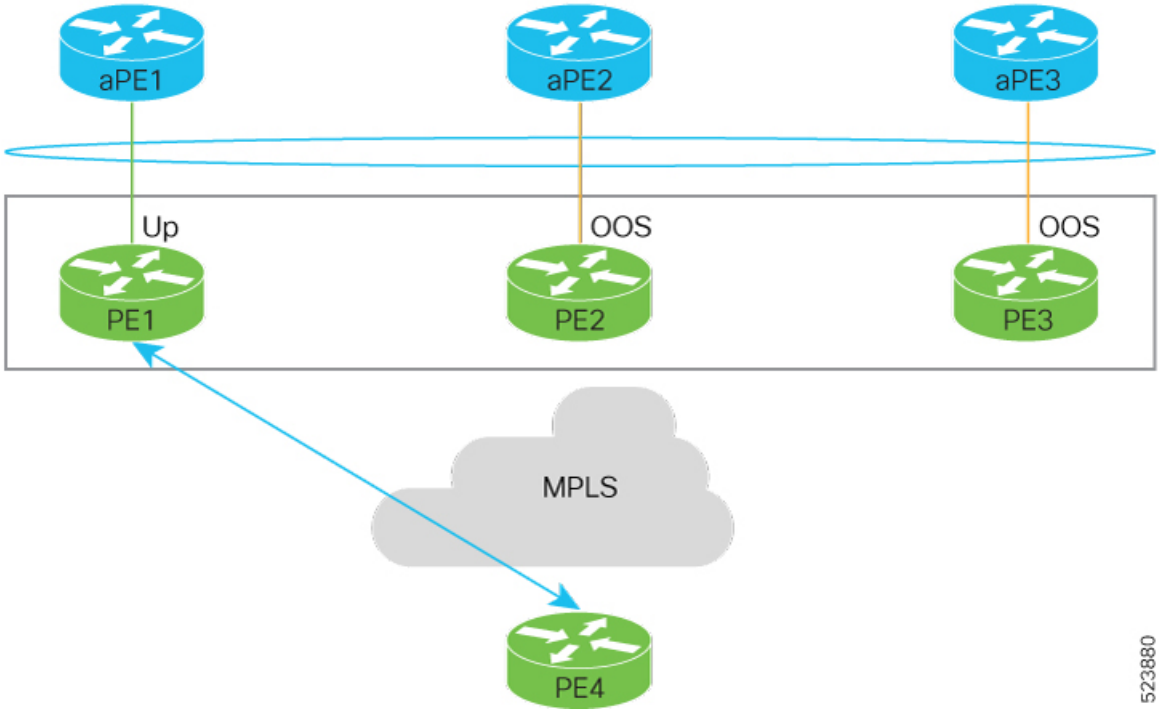
- The feature is supported only on EVPN PEs in the port-active mode.
- The bundle attached to the ethernet segment must be configured with the **lacp mode active** command.

  The **LACP mode on** command is not supported.

Topology

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

Figure 14: EVPN DF Election

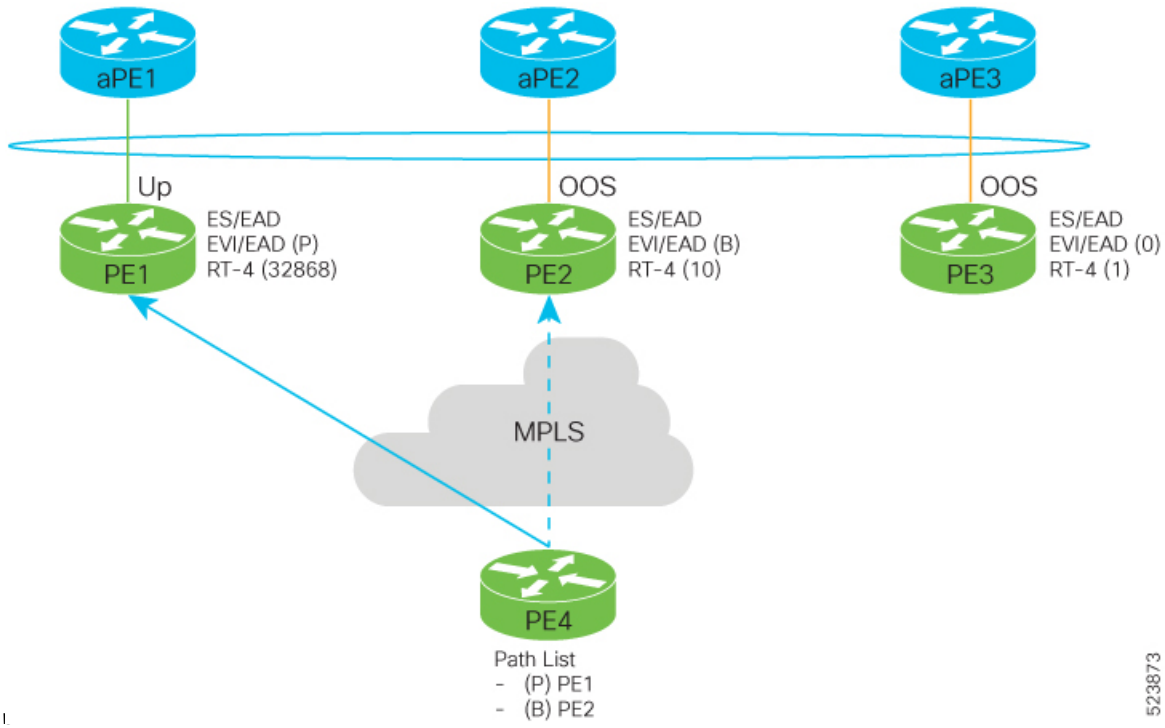


- PE1, PE2, and PE3 are PEs for the EVPN core network.
- aPE1, aPE2, and aPE3 are their access PE counterparts and configured in a multichassis link aggregation group (MCLAG) redundancy group. Only one link among the three is active at any given time. aPE1, aPE2, and aPE3 are in a non-revertive mode.
- PE1 is directly connected to aPE1, PE2 to aPE2, and PE3 to aPE3. EVPN VPWS is configured on the PE devices in the core.
- All PE devices are attached to the same bundle and shares the same ethernet segment identifier.
- PE1, PE2, and PE3 are configured with a weight of 100, 10, and 1 respectively.

Traffic Flow

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

Figure 15: Traffic



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

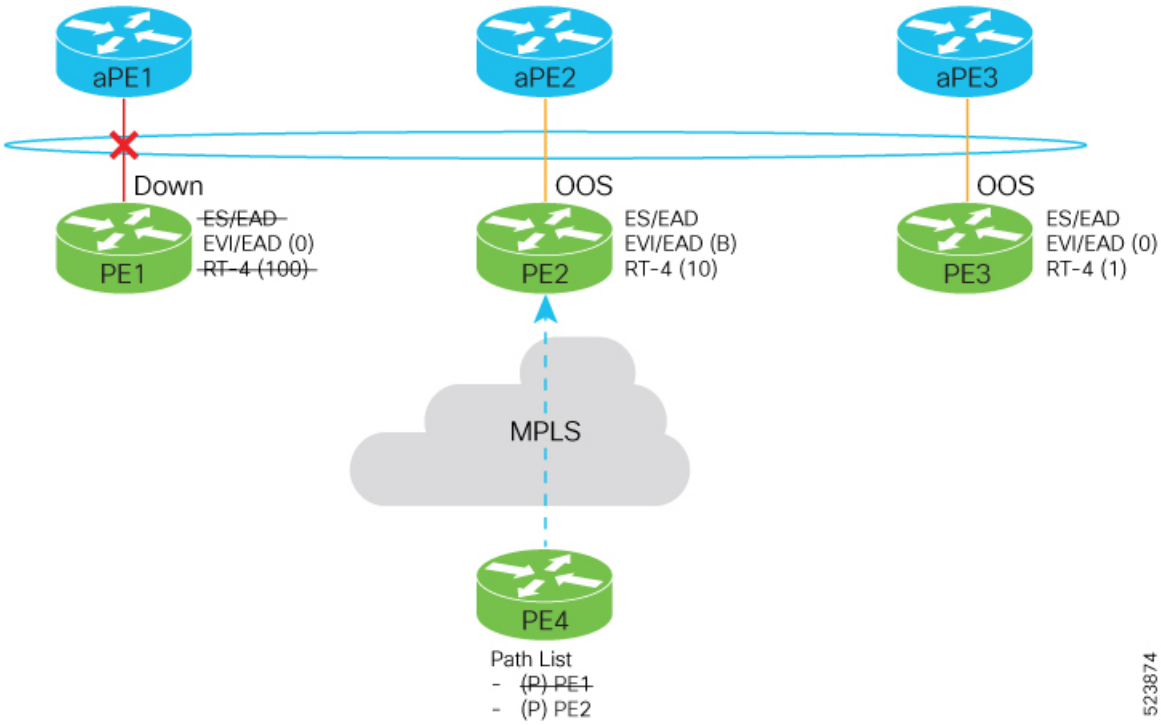
Failure and Recovery Scenarios

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

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

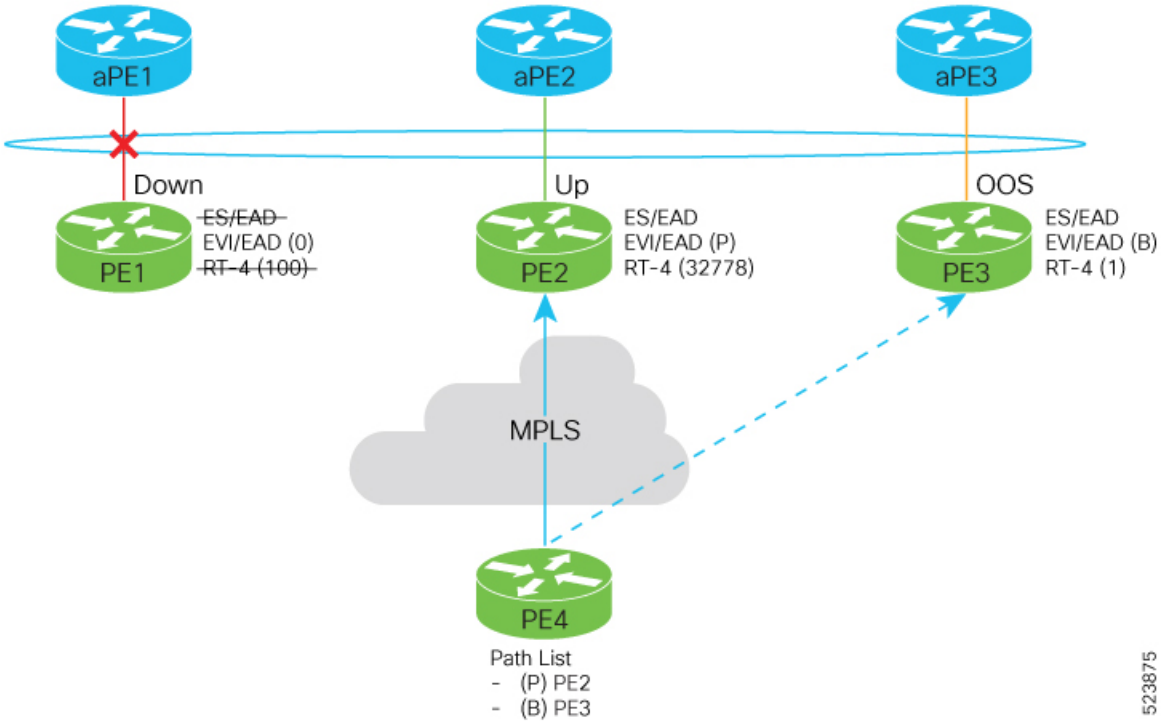
Scenario - 1

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



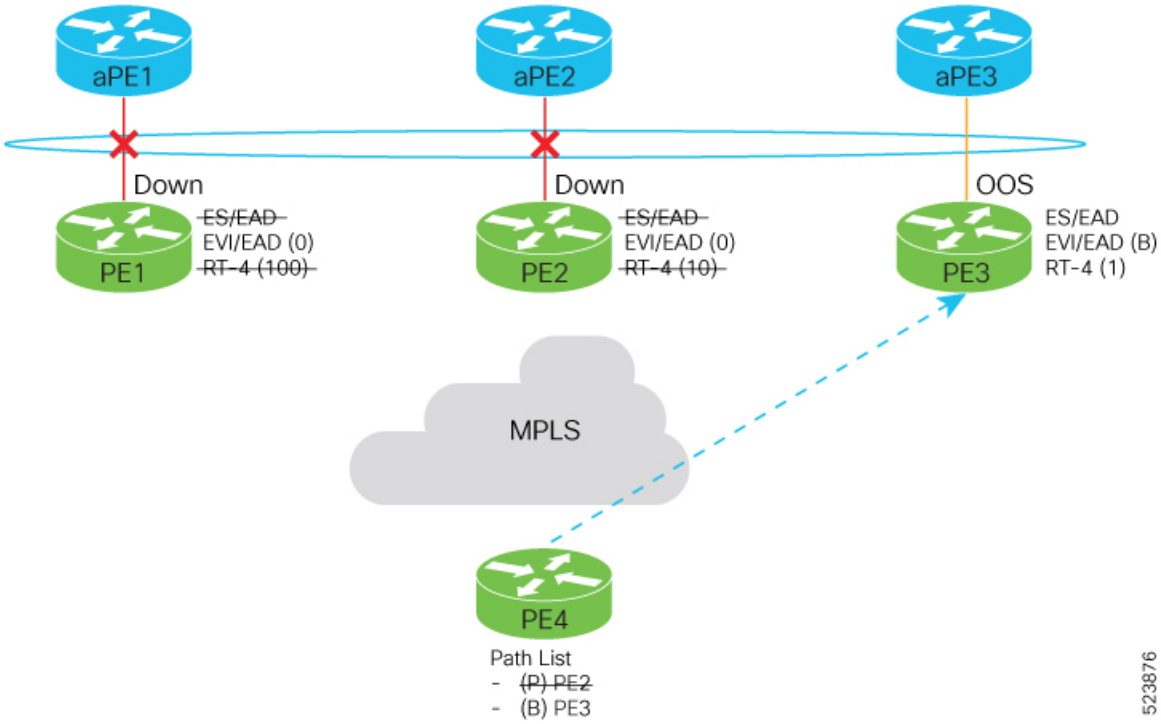
When aPE1-PE1 interface is down, the PE1 withdraws the EAD/ES route, and the traffic is sent through the backup path, which is PE2.

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

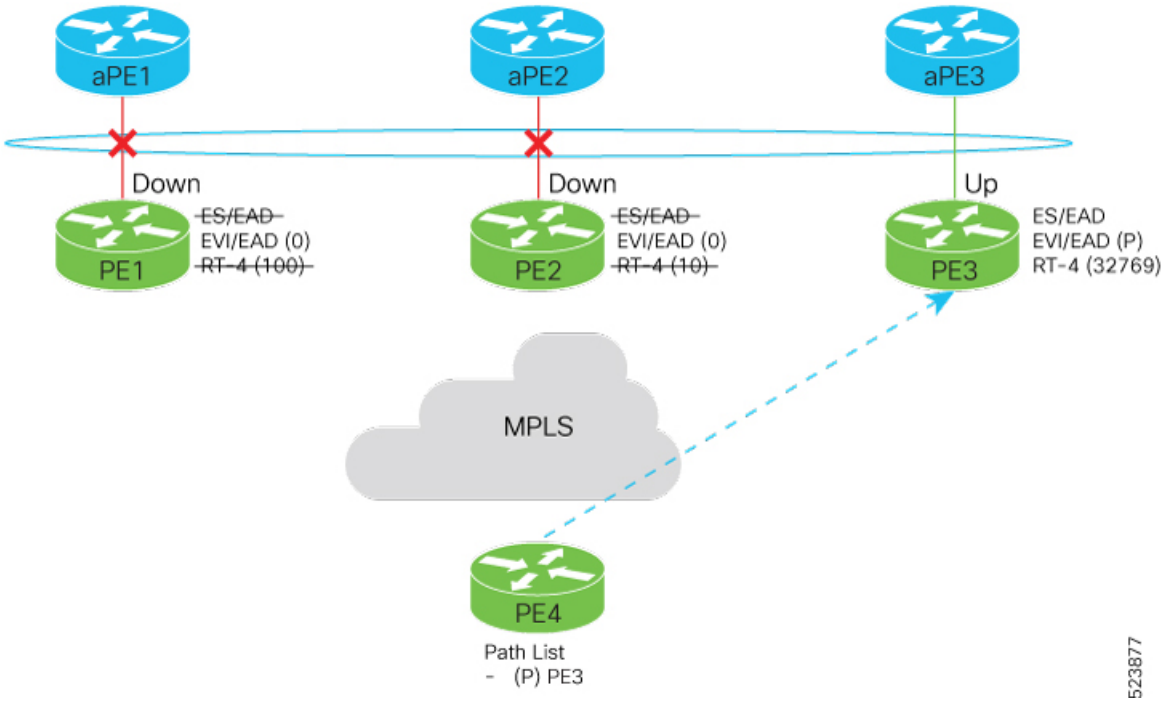


Scenario - 2

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

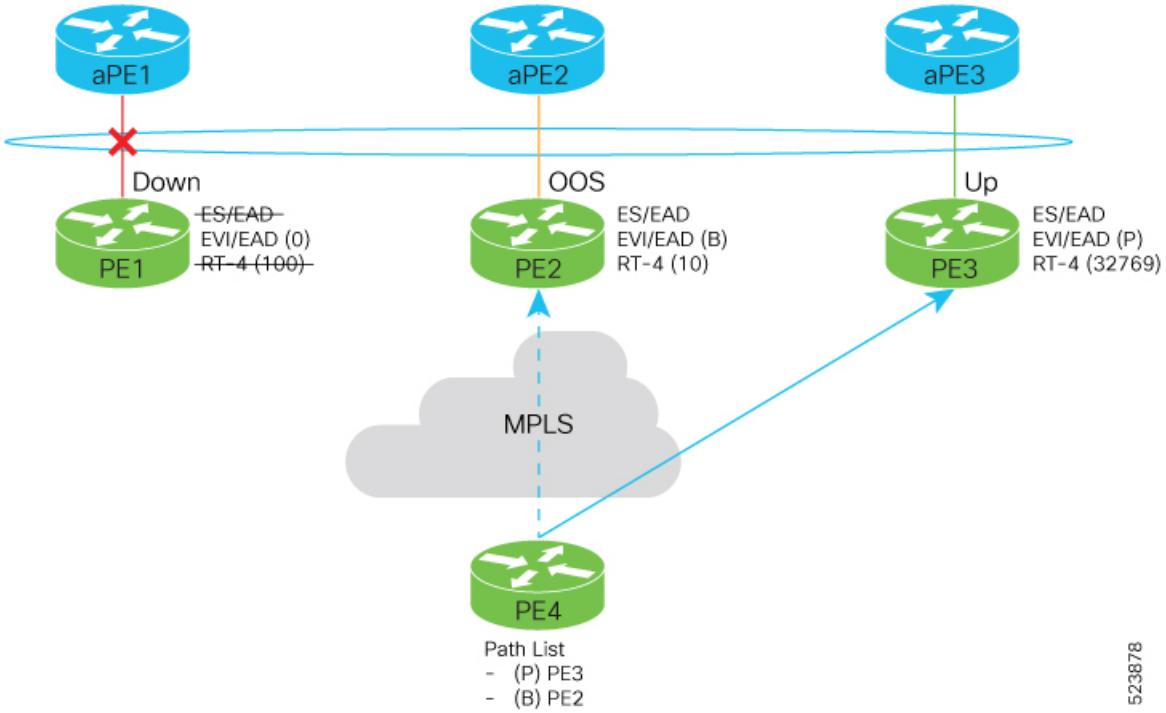


When the aPE2-PE2 interface is also down, the traffic is sent through aPE3-PE3 link. aPE3-PE3 becomes the primary path with a weight of 32769.



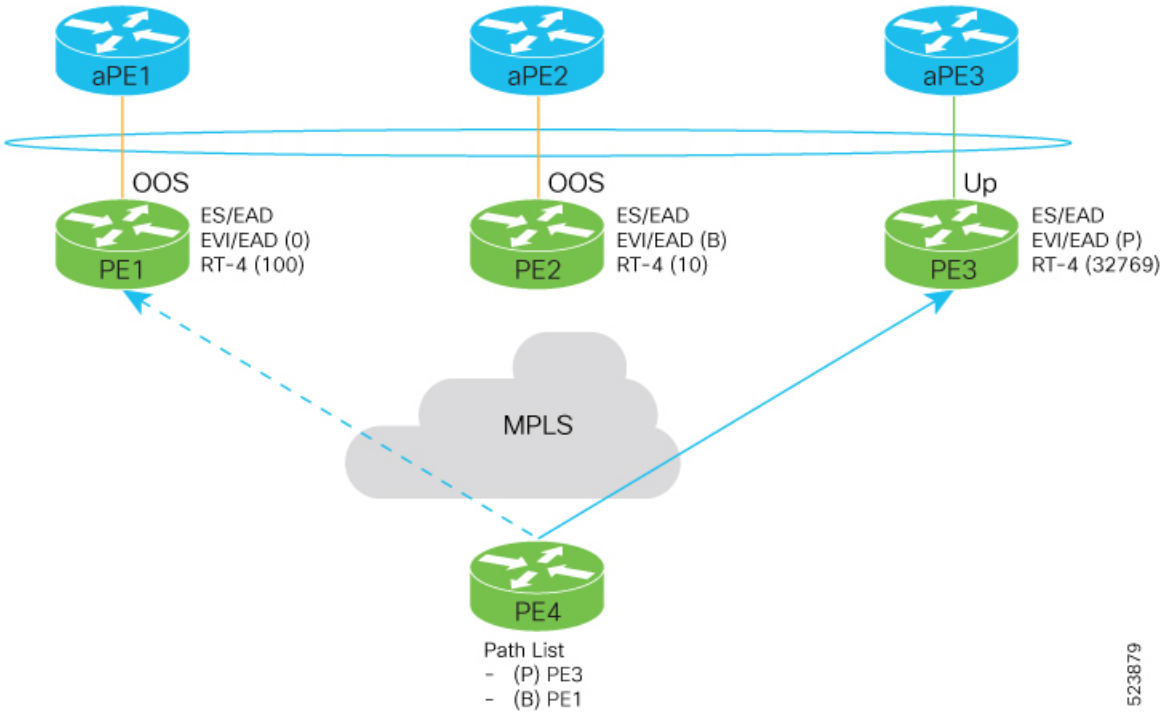
Scenario - 3

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



Scenario - 4

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



Configure EVPN DF Election

Perform the following tasks to configure access-driven and preference based EVPN DF Election:

- Configure EVPN DF election on PE1, PE2, and PE3, with the service carving mode as preference-based and access-driven.
- Configure LACP on aPE1, aPE2, and aPE3

Configuration Example

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

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

/*  PE1 Configuration */
Router#configure
Router(config)#evpn
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment
Router(config-evpn-ac-es)#identifier type 0 01.11.00.00.00.00.00.00.01
Router(config-evpn-ac-es)#load-balancing-mode port-active
```

```
Router(config-evpn-ac-es)#service-carving preference-based
Router(config-evpn-ac-es-sc-pref)#weight 100
Router(config-evpn-ac-es-sc-pref)#access-driven
Router(config-evpn-ac-es-sc-pref)#commit

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

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

Configure LACP on aPE1, aPE2, and aPE3

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

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

/* aPE3 Configuration */
Router#configure
Router(config)#interface Bundle-Ether 1
Router(config-if)#lacp non-revertive
Router(config-if)#bundle maximum-active links 1 hot-standby
Router(config-if)#exit
```

```
Router(config-if)#interface GigabitEthernet0/0/0/38
Router(config-if)bundle id 10 mode active
Router(config-if)bundle port-priority 30000
Router(config-if)description Connection to PE3
Router(config-if)commit
```

Running Configuration

```
/* PE1 Configuration */
evpn
 interface Bundle-Ether 1
  ethernet-segment
   identifier type 0 01.11.00.00.00.00.00.01
    load-balancing-mode port-active
    service-carving preference-based
    weight 100
    access-driven
   !
 !

/* PE2 Configuration */
evpn
 interface Bundle-Ether 1
  ethernet-segment
   identifier type 0 01.11.00.00.00.00.00.01
    load-balancing-mode port-active
    service-carving preference-based
    weight 10
    access-driven
   !
 !

/* PE3 Configuration */
evpn
 interface Bundle-Ether 1
  ethernet-segment
   identifier type 0 01.11.00.00.00.00.00.01
    load-balancing-mode port-active
    service-carving preference-based
    weight 1
    access-driven
   !
 !

/* aPE1 Configuration */

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

/* aPE2 Configuration */

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

```
 description Connection to PE2
!

/* aPE3 Configuration */

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

Verification

The following output shows configuration of the EVPN DF Election.

```
Router#show evpn ethernet-segment detail
Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0001.0001.0001.1b01.001b BE1                               192.168.0.1
                                                           192.168.0.3
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether1
     Interface MAC  : 02ef.af8d.8008
     IfHandle       : 0x00004190
     State          : Up
     Redundancy     : Active
  ESI type          : 0
     Value          : 01.0001.0001.1b01.001b
  ES Import RT      : 0100.0100.011b (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH
     Configured     : Port-Active
  Service Carving   : Preferential
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 2 Nexthops
     192.168.0.1 [PREF:P:d6ce:T] >> Weight in hexadecimal
     192.168.0.3 [PREF:P:457]
  Service Carving Synchronization:
     Mode           : NONE
     Peer Updates   :
  Service Carving Results:
     Forwarders     : 3
     Elected        : 3
     Not Elected    : 0
  EVPN-VPWS Service Carving Results:
     Primary        : 1
     Backup         : 0
     Non-DF         : 0
  MAC Flushing mode : STP-TCN
  Peering timer     : 3 sec [not running]
  Recovery timer    : 30 sec [not running]
  Carving timer     : 0 sec [not running]
  Local SHG label   : 28384
  Remote SHG labels : 0
  Access signal mode: Bundle OOS (Default)
```

Highest Random Weight Mode for EVPN DF Election

The Highest Random Weight (HRW) Mode for EVPN DF Election feature provides optimal load distribution of Designated Forwarder (DF) election, redundancy, and fast access. It ensures a nondisruptive service for an ES irrespective of the state of a peer DF.

The DF election is calculated based on the weight. The highest weight becomes the DF and the subsequent weight becomes a backup DF (BDF). The weight is determined by the mathematical function of EVI, ESI, and the IP address of the server.

DF weight calculation is based on the weight vector:

```
Wrand(v, Si) = (1103515245((1103515245.Si+12345)XOR
       D(v))+12345)(mod 2^31)
                     where:
             Si: IP Address of the server i
             v: EVI
             D(v): 31 bit digest [CRC-32 of v]
```

The existing DF election algorithm is based on ordinal value of a modulus calculation, and it comprises of number of peers and EVI. The DF is determined by the mathematical function of ESI and EVI, which is called “service carving”. This mode of DF election is described in RFC 7432.

In modulus calculation mode, the algorithm does not perform well when the Ethernet tags are all even or all odd. When the Ethernet Segment (ES) is multihomed to two PEs, all the VLANs pick only one of the PEs as the DF; one of the PEs does not get elected at all as the DF. The DF election is not optimal in this mode of operation.

The HRW mode of DF election has the following advantages over modulus mode of DF election:

- The DF election for the respective VLANs is equally distributed among the PEs.
- When a PE which is neither a DF nor a BDF hosts some VLANs on a given ES, and if the PE goes down, or its connection to the ES goes down, it does not result in a DF and BDF reassignment to the other PEs. This eliminates computation during the connection flaps.
- It avoids the service disruption that are inherent in the existing modulus based algorithm.
- The BDF provides redundant connectivity. The BDF ensures that there is no traffic disruption when a DF fails. When a DF fails, the BDF becomes the DF.

Configure Highest Random Weight Mode for EVPN DF Election

Perform this task to configure Highest Random Weight Mode for EVPN DF Election feature.

Configuration Example

```
Router# configure
Router(config)#evpn
Router(config-evpn)#interface Bundle-Ether 23
Router(config-evpn-ac)#ethernet-segment
Router(config-evpn-ac-es)#service-carving hrw
Router(config-evpn-ac-es)#commit
```

Running Configuration

```
configure
 evpn
  interface Bundle-Ether 23
   ethernet-segment
    service-carving hrw
   !
  !
 !
```

Verification

Verify that you have configured HRW mode of DF election.

```
Router#show evpn ethernet-segment interface bundleEther 23 carving detail
Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0011.1111.1111.1111.1111 Gi0/2/0/0                         192.168.0.2
                                                           192.168.0.3
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : GigabitEthernet0/2/0/0
     Interface MAC  : 02db.c740.ca4e
     IfHandle       : 0x01000060
     State          : Up
     Redundancy     : Not Defined
  ESI type          : 0
     Value          : 11.1111.1111.1111.1111
  ES Import RT      : 0011.0011.0011 (Local)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH, Single-active
     Configured     : Single-active (AApS) (default)
  Service Carving   : HRW    -> Operation mode of carving
  Peering Details   : 192.168.0.2[HRW:P:00] 192.168.0.3[HRW:P:00] -> Carving capability as
 advertised by peers
  Service Carving Results:
     Forwarders     : 1
     Permanent      : 0
     Elected        : 0
     Not Elected    : 1
  MAC Flushing mode : STP-TCN
  Peering timer     : 3 sec [not running]
  Recovery timer    : 30 sec [not running]
  Carving timer     : 0 sec [not running]
  Local SHG label   : 28109
  Remote SHG labels : 1
              24016 : nexthop 192.168.0.3
```

Virtual Ethernet Segment

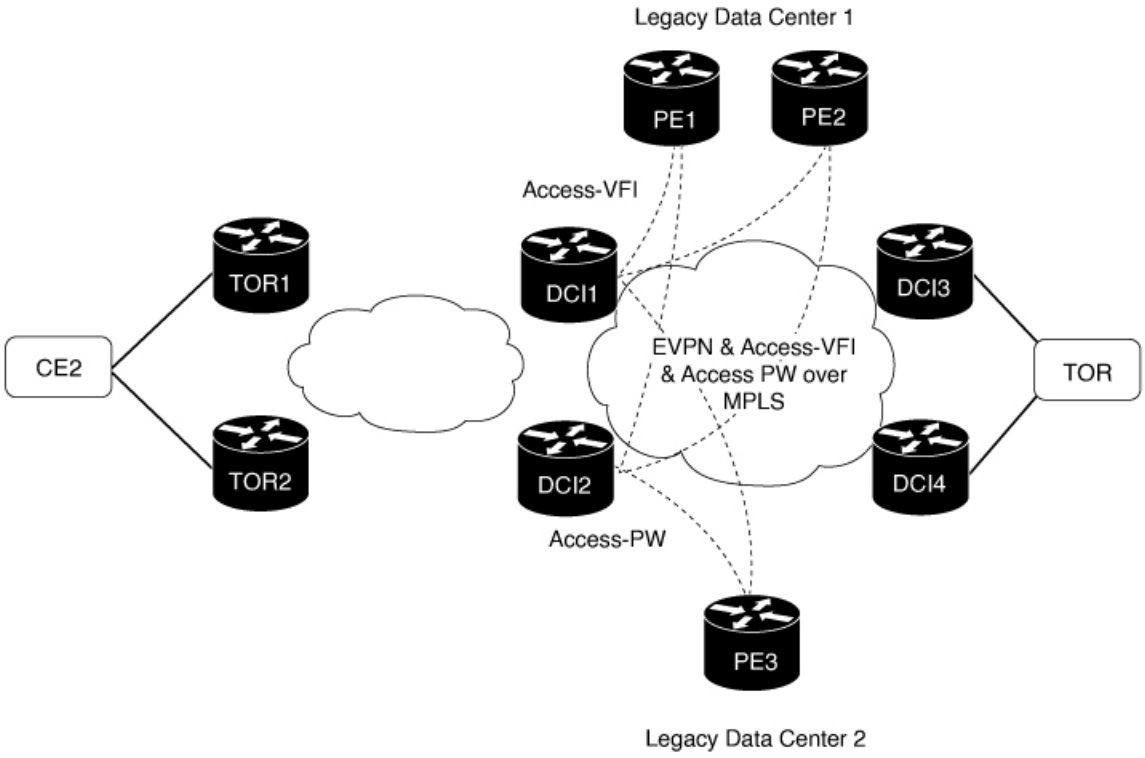
Table 13: Feature History Table

Feature Name	Release Information	Feature Description

Virtual Ethernet Segment on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The Virtual Ethernet Segment is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Virtual Ethernet Segment	Release 24.2.11	A Virtual Ethernet Segment (VES) allows a Customer Edge (CE) device to connect to an EVPN service over an MPLS network, which can be used for redundancy and load balancing. This feature is supported only on routers with the 88-LC1-36EH line cards.

Traditionally, multi-homing access to EVPN bridge is through bundle Ethernet connection or a physical Ethernet connection. A customer edge (CE) is connected to multiple provider edges (PEs) and each CE-PE pair is an Ethernet segment (ES). When multiple Ethernet segments are made to appear as one common Ethernet segment to the CE device, it is a Virtual Ethernet Segment (VES). The VES allows a CE to access EVPN bridge through MPLS network. The logical connection between CE and PE is a pseudowire (PW). You can use VES to access PW and AC sub-interface, which is used for redundancy and load balancing.

Figure 16: Virtual Ethernet Segment (VES)



Consider the topology where EVPN data centers, DCI1 and DCI2 are connected to legacy data centers through access PW on a single Ethernet segment, which is VES. In the topology, the traffic flow from CE2 reaches the legacy data centers through EVPN data centers using VES.

Consider a traffic flow from CE2 to PE3, the Legacy Data Center 2.

- CE2 sends the traffic to DCI1 or DCI2 through EVPN.
- DCI1 and DCI2 are connected to PE3 through access PW on a single Ethernet segment.
- DCI1 and DCI2 advertise Type 4 routes, and then perform designated forwarder (DF) election after they discover each other. One of them becomes a DF and other a non-DF.
- The traffic is forwarded through the DF. The non-DF path is in standby mode. DCI1 or DCI2, whichever is the DF, sends the traffic to PE3.

Consider a traffic flow from CE2 to PE1 and PE2, the Legacy Data Center 1.

- CE2 sends the traffic to DCI1 or DCI2 through EVPN.
- DCI1 or DCI2 sends the traffic to PE1 and PE2.
- DCI1 and DCI2 advertise Type 4 routes, and then perform DF election after they discover each other. One of them becomes a DF and other a non-DF.
- One of them becomes a DF and other a non-DF.
- The traffic is forwarded through the DF. The non-DF path is in standby mode. DCI1 or DCI2, whichever is the DF, sends the traffic to PE1 and PE2.

Configure Virtual Ethernet Segment

The following section describes how to configure access PW that acts as VES.

1. Configure DCI1 and DCI2 with bridge domain and assign EVI to the bridge domain.
2. Configure EVPN with virtual ethernet segment on both DCI1 and DCI2.
3. Configure PE3 with bridge domain and assign the virtual ethernet segments of DC1 and DCI2 as neighbors to the bridge domain.

```
/* Configure DCI1 */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group bg1
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain bd1
RP/0/RSP0/CPU0:router(config-bg-bd)# neighbor 70.70.70.70 pw-id 17300001
RP/0/RSP0/CPU0:router(config-bg-bd-pw)# evi 1
RP/0/RSP0/CPU0:router(config-bg-bd-pw-evi)# member vni 10001

/* Configure EVPN */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# virtual neighbor 70.70.70.70 pw-id 17300001
RP/0/RSP0/CPU0:router(config-evpn-ac-pw)# ethernet-segment
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-es)# identifier type 0 12.12.00.00.00.01.00.00.03
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-es)# bgp route-target 1212.8888.0003
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-es)# exit
RP/0/RSP0/CPU0:router(config-evpn-ac-pw)# timers peering 15
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-timers)# commit

/* Configure DCI2 */
RP/0/RSP0/CPU0:router# configure
```

```
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group bg1
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain bd1
RP/0/RSP0/CPU0:router(config-bg-bd)# neighbor 70.70.70.70 pw-id 27300001
RP/0/RSP0/CPU0:router(config-bg-bd-pw)# evi 1
RP/0/RSP0/CPU0:router(config-bg-bd-pw-evi)# member vni 10001

/* Configure EVPN */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# virtual neighbor 70.70.70.70 pw-id 27300001
RP/0/RSP0/CPU0:router(config-evpn-ac-pw)# ethernet-segment
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-es)# identifier type 0 12.12.00.00.00.01.00.00.03
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-es)# bgp route-target 1212.8888.0003
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-es)# exit
RP/0/RSP0/CPU0:router(config-evpn-ac-pw)# timers peering 15
RP/0/RSP0/CPU0:router(config-evpn-ac-pw-timers)# commit

/* Configure PE3 */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group 73
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain 73-1
RP/0/RSP0/CPU0:router(config-bg-bd)# neighbor 10.10.10.10 pw-id 17300001
RP/0/RSP0/CPU0:router(config-bg-bd-pw)# exit
RP/0/RSP0/CPU0:router(config-bg-bd)# neighbor 20.20.20.20 pw-id 27300001
RP/0/RSP0/CPU0:router(config-bg-bd)# commit
```

Running Configuration

This section shows access PW running configuration.

```
/* On DCI1 */

l2vpn
 bridge group bg1
  bridge-domain bd1
   neighbor 70.70.70.70 pw-id 17300001
    evi 1
     member vni 10001
!

evpn
  virtual neighbor 70.70.70.70 pw-id 17300001
   ethernet-segment
    identifier type 0 12.12.00.00.00.01.00.00.03
    bgp route-target 1212.8888.0003
    !
  timers peering 15
!

/* On DCI2 */

l2vpn
 bridge group bg1
  bridge-domain bd1
   neighbor 70.70.70.70 pw-id 27300001
    evi 1
     member vni 10001
!

evpn
  virtual neighbor 70.70.70.70 pw-id 27300001
```

```
   ethernet-segment
    identifier type 0 12.12.00.00.00.01.00.00.03
    bgp route-target 1212.8888.0003
    !
  timers peering 15
!

/* On PE3 */
!
l2vpn
 bridge group bg73
  bridge-domain bd73-1
   neighbor 10.10.10.10 pw-id 17300001
   !
   neighbor 20.20.20.20 pw-id 27300001

!
```

Verification

The following output shows the virtual access PW configuration.

```
Router# show evpn ethernet-segment
Thu Mar  7 10:56:37.662 UTC

Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0012.1200.0000.0100.0003 PW:70.70.70.70,17300001           N/A
RP/0/RP0/CPU0:ios#show evpn ethernet-segment  detail
Thu Mar  7 10:56:53.806 UTC
Legend:
  B   - No Forwarders EVPN-enabled,
  C   - MAC missing (Backbone S-MAC PBB-EVPN / Grouping ES-MAC vES),
  RT  - ES-Import Route Target missing,
  E   - ESI missing,
  H   - Interface handle missing,
  I   - Name (Interface or Virtual Access) missing,
  M   - Interface in Down state,
  O   - BGP End of Download missing,
  P   - Interface already Access Protected,
  Pf  - Interface forced single-homed,
  R   - BGP RID not received,
  S   - Interface in redundancy standby state,
  X   - ESI-extracted MAC Conflict
  SHG - No local split-horizon-group label allocated
  Hp  - Interface blocked on peering complete during HA event
  Rc  - Recovery timer running during peering sequence

Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0012.1200.0000.0100.0003 PW:70.70.70.70,17300001           N/A
  ES to BGP Gates   : R
  ES to L2FIB Gates : Ready
  Virtual Access    :
     Name           : PW_70.70.70.70_17300001
     State          : Peering
     Num PW Up      : 0
  ESI ID            : 1
  ESI type          : 0
     Value          : 0012.1200.0000.0100.0003
  ES Import RT      : 1212.8888.0003 (Local)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : SH
```

```
Configured     : Single-active (AApS) (default)
```

EVPN Cost-Out

Table 14: Feature History Table

Feature Name	Release Information	Feature Description
EVPN Cost-Out on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The EVPN Cost-Out is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN Cost-Out	Release 24.2.11	The cost-out node brings down the bundle interfaces on the PE to prepare the node for reload or software upgrade. By costing out a node, the traffic is steered away from the PE without any traffic disruption. This allows you to manage the network traffic effectively while reloading or upgrading a node. This feature is supported only on routers with the 88-LC1-36EH line cards.

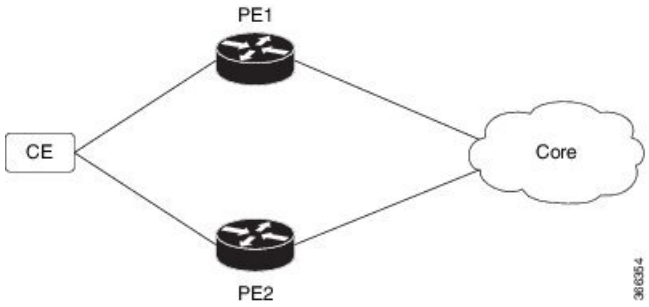
EVPN cost-out enables you to control the state of bundle interfaces that are part of an Ethernet segment that have Link Aggregation Control protocol (LACP) configured. This feature enables you to put a node out of service (OOS) without having to manually shutdown all the bundles on their provider edge (PE) and prepare the node for reload or software upgrade.

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

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

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

Figure 17: EVPN Cost-Out



To bring up the node into service, use the **no cost-out** command. This brings up all the bundle interfaces belonging to EVPN Ethernet segment on the PE and the corresponding bundle members on the CE.

When the node is in cost-out state, adding a new bundle Ethernet segment brings that bundle down. Similarly, removing the bundle Ethernet segment brings that bundle up.

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

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

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

Configure EVPN Cost-Out

The following examples show configuration of cost-out and startup cost-in timer.

```
/* Configuring cost-out to bring down the node on a PE */

Router# configure
Router(config)# evpn
Router(config-evpn)# cost-out
Router(config-evpn)commit

/* Bringing up the node into service */

Router# configure
Router(config)# evpn
Router(config-evpn)# no cost-out
Router(config-evpn)commit

/* Configuring the timer to bring up the node into service after the specified time on
reload */

Router# configure
Router(config)# evpn
Router(config-evpn)# startup-cost-in 6000
Router(config-evpn)commit
```

Running Configuration

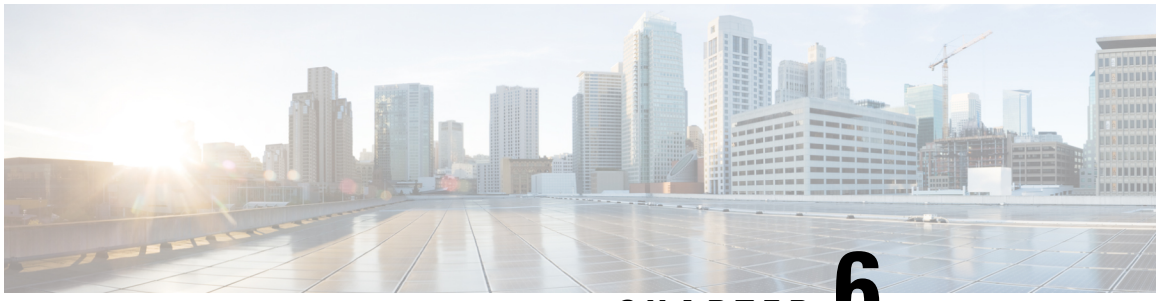
```
configure
evpn
 cost-out
!

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

Verification

The following examples show outputs to verify the cost-out and startup cost-in timer configurations.

```
/* Verify the node cost-out configuration */

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


/* Verify the no cost-out configuration */

Router# show evpn summary
Fri Apr  7 07:45:22.311 IST
Global Information
----------------------------
Number of EVIs                : 2
Number of Local EAD Entries   : 0
Number of Remote EAD Entries  : 0
Number of Local MAC Routes    : 0
Number of Local MAC Routes    : 5
          MAC                 : 5
```

```
          MAC-IPv4            : 0
          MAC-IPv6            : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes   : 7
          MAC                 : 7
          MAC-IPv4            : 0
          MAC-IPv6            : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels     : 5
Number of ES Entries          : 9
Number of Neighbor Entries    : 1
EVPN Router ID                : 192.168.0.1
BGP Router ID                 : ::
BGP ASN                       : 100
PBB BSA MAC address           : 0207.1fee.be00
Global peering timer          :      3 seconds
Global recovery timer         :     30 seconds
EVPN cost-out                 : FALSE
      startup-cost-in timer   : Not configured


/* Verify the startup-cost-in timer configuration */

Router# show evpn summary
Fri Apr  7 07:45:22.311 IST
Global Information
----------------------------
Number of EVIs                : 2
Number of Local EAD Entries   : 0
Number of Remote EAD Entries  : 0
Number of Local MAC Routes    : 0
Number of Local MAC Routes    : 5
          MAC                 : 5
          MAC-IPv4            : 0
          MAC-IPv6            : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes   : 7
          MAC                 : 7
          MAC-IPv4            : 0
          MAC-IPv6            : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels     : 5
Number of ES Entries          : 9
Number of Neighbor Entries    : 1
EVPN Router ID                : 192.168.0.1
BGP Router ID                 : ::
BGP ASN                       : 100
PBB BSA MAC address           : 0207.1fee.be00
Global peering timer          :      3 seconds
Global recovery timer         :     30 seconds
EVPN node cost-out            : TRUE
        startup-cost-in timer : 6000
```



CHAPTER 6

EVPN Features

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

BUM Ingress Replication for EVPN E-LAN

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
BUM Ingress Replication for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The BUM ingress replication is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
BUM Ingress Replication for EVPN E-LAN on the 88-LC1-36EH Line Cards	Release 24.2.11	The BUM ingress replication is now supported on routers with the 88-LC1-36EH line cards.

Feature Name	Release Information	Feature Description
BUM Ingress Replication for EVPN E-LAN	Release 7.11.1	You can optimize Broadcast, Unknown Unicast, and Multicast (BUM) traffic by ensuring that traffic that a 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 E-LAN

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
Split-Horizon Groups for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The split-horizon groups for EVPN E-LAN is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Split-Horizon Groups for EVPN E-LAN on the 88-LC1-36EH Line Cards	Release 24.2.11	The split-horizon groups for EVPN E-LAN is now supported on routers with the 88-LC1-36EH line cards.
Split-Horizon Groups for EVPN E-LAN	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 17: 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 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 E-LAN

Table 18: Feature History Table

Feature Name	Release Information	Feature Description
VRF Leaking for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The VRF leaking for EVPN E-LAN is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
VRF Leaking for EVPN E-LAN on the 88-LC1-36EH Line Cards	Release 24.2.11	The VRF leaking for EVPN E-LAN is now supported on routers with the 88-LC1-36EH line cards.
VRF Leaking for EVPN E-LAN	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 network, 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 network. 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 E-LAN

Table 19: Feature History Table

Feature Name	Release Information	Feature Description
Core Isolation by Interface Tracking on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The core isolation by interface tracking is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Core Isolation by Interface Tracking on the 88-LC1-36EH Line Cards	Release 24.2.11	The core isolation by interface tracking is now supported on routers with the 88-LC1-36EH line cards.
Core Isolation by Interface Tracking for EVPN E-LAN	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.

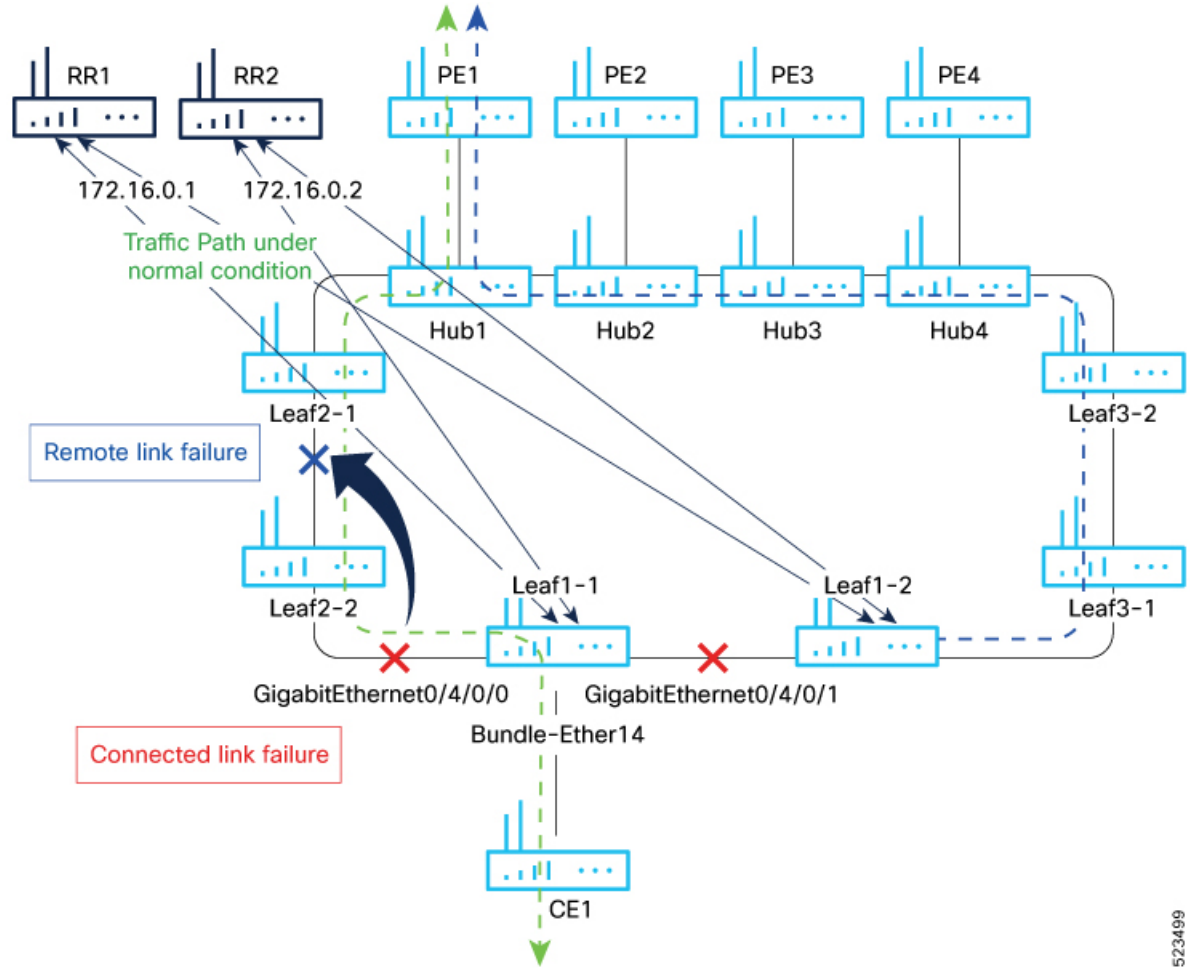
When the state of the tracked object changes, the tracking process receives the notification. The tracked objects can have an up or down state. A list can track multiple objects using a flexible method that combines objects with Boolean logic. This functionality includes:

- 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 18: 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 address-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 default reachability
  Up
neighbor-B                      bgp nbr L2VPN EVPN 172.16.0.1 vrf default 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 20: Feature History Table

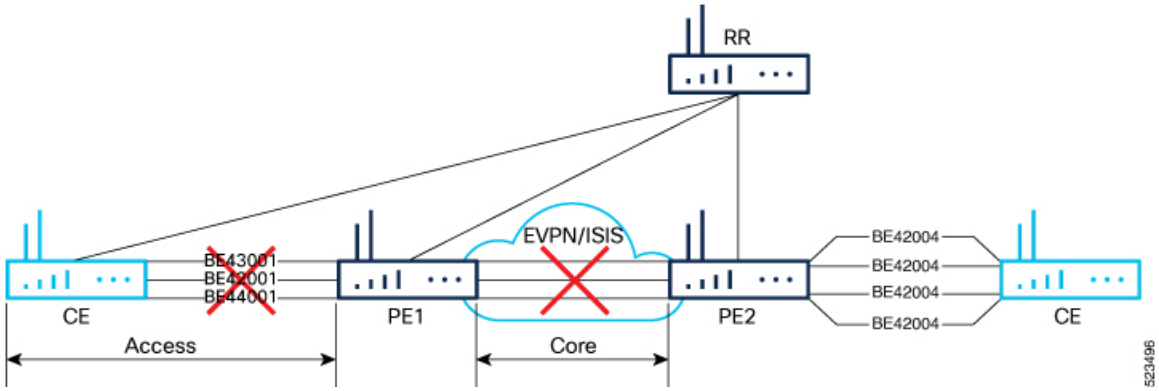
Feature Name	Release Information	Feature Description
EVPN Core Isolation through Peer Failure Detection on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	EVPN Core Isolation through Peer Failure Detection is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN Core Isolation through Peer Failure Detection on the 88-LC1-36EH Line Cards	Release 24.2.11	EVPN Core Isolation through Peer Failure Detection is now supported on the 88-LC1-36EH line cards.
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 19: 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 E-LAN

Table 21: Feature History Table

Feature Name	Release Information	Feature Description
MAC Mobility for EVPN E-LAN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The MAC mobility for EVPN E-LAN is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
MAC Mobility for EVPN E-LAN on the 88-LC1-36EH Line Cards	Release 24.2.11	The MAC mobility for EVPN E-LAN is now supported on routers with the 88-LC1-36EH line cards.
MAC Mobility for EVPN E-LAN	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 22: Feature History Table

Feature Name	Release Information	Feature Description
Detect and Block Duplicate MAC Addresses on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The Detect and Block Duplicate MAC Addresses feature is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Detect and Block Duplicate MAC Addresses on the 88-LC1-36EH Line Cards	Release 24.2.11	The Detect and Block Duplicate MAC Addresses feature is now supported on the 88-LC1-36EH line cards.

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 for EVPN E-LAN

Table 23: Feature History Table

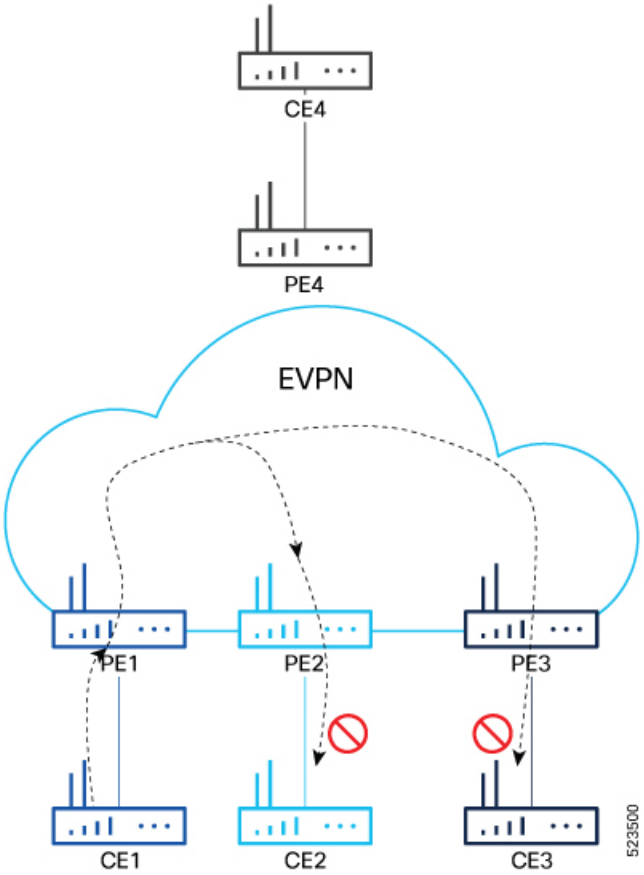
Feature Name	Release Information	Feature Description
EVPN E-Tree (Scenario 2) on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	EVPN E-Tree (Scenario 2) is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN E-Tree (Scenario 1a) on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	EVPN E-Tree (Scenario 1a) is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN E-Tree (Scenario 2)	Release 24.2.11	We now enable a PE device to have both root and leaf sites for a given EVI, which increases the granularity of leaf designation from the entire bridge to AC bridge ports; ACs under a bridge may be root or leaf. This feature is supported on routers with the 88-LC1-36EH line cards.
EVPN E-Tree (Scenario 1a)	Release 24.2.11	We now support EVPN E-Tree with route-targets (RT) constraints using two RTs per EVI on routers with the 88-LC1-36EH line cards.
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 20: 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 and Scenario 2 are 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.

Scenario 2

Customer sites represented by ACs can be defined as root or leaf. A PE can receive traffic from both root and leaf ACs for a given EVI from a remote node. An EVI can be associated with both roots and leafs. If an AC is not configured as E-Tree leaf, it is considered as root by default.

In Scenario 2, a PE for a given EVI can have both root and leaf sites. The granularity of root or leaf designation increases from bridge-domain level in Scenario 1 to per AC level. Traffic for an EVI from a PE in the network can be from either a root or a leaf site.

Scenario 2 Behavior for Unicast Traffic:

- Remote PE performs ingress filtering to prevent traffic from being sent unnecessarily over the core to be filtered at egress PE.
- PE indicates if the MAC address is associated with a root or a leaf.
- MAC advertisements originating from a leaf site are colored with a leaf-indication flag in an extended community; routes that do not have this flag are from a root site.
- Remote PEs perform ingress filtering, when MAC is programmed and the leaf-indication is present, a cross-check is performed with the originating AC; if the AC is also a leaf, packets will not be forwarded.
- Supports E-Tree extcomm of type 0x06 (EVPN) and sub-type 0x05 used for leaf-indication for known-unicast and BUM traffic.
- Enable unknown unicast suppression at EVIs connected to both root and leaf sites to prevent egress unknown unicast traffic arriving at an EVI from being flooded to ACs. This eliminates the leaf-to-leaf traffic during a BD MAC flush.
- MAC advertises local ESI and there is no leaf indicator from root.

  When processing the root sync-route, the root or leaf status of the individual AC is considered, instead of the entire bridge-domain. If a root MAC with a matching local ESI is received, and if the corresponding AC is a leaf, a syslog message is generated for the misconfiguration.

Scenario 2 Behavior for BUM Traffic:

- The PE performs egress filtering for BUM traffic. BUM traffic originating from leaf sites is filtered at egress nodes if the destination is also a leaf.
- A PE with leaf sites allocate a leaf label, and communicate this label to remote PEs using an ES/EAD route with ESI 0 with the ETREE extended community.
- BUM traffic originating from single-homed leaf AC is tagged with destination etree leaf label.
- BUM traffic originating from single-homed root AC is not tagged with any ESI or etree leaf label.
- BUM traffic originating from multi-homed leaf AC is tagged with destination etree leaf label.
- BUM traffic originating from multi-homed root AC is tagged with ESI label.
- The ingress PE tags MPLS frames with the etree leaf label for traffic originating from a leaf site; this label allows the disposition PE to perform egress filtering to native EVPN ESI label filtering.
- Intra-PE forwarding between leaf sites is prevented by putting all leaf ACs under a given bridge domain in a single split-horizon group.

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
```

Configure EVPN E-Tree (Scenario 2)

Perform the following tasks on the PE device where you want to configure E-tree (Scenario 2):

1. Configure bridge domain
2. Configure E-Tree for the root and leaf
3. Configure EVI

Configuration Example

The following example shows configuration of EVPN E-Tree for root and leaf:

```
/* Configure bridge domain */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd_1

/* Configure E-Tree */
Router(config-l2vpn-bg-bd)# interface Bundle-Ether400
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# # interface Bundle-Ether401.1001
Router(config-l2vpn-bg-bd-ac)# etree
Router(config-l2vpn-bg-bd-ac-etree)# leaf

/* Configure EVI */
Router(config-l2vpn-bg-bd)# evi 200
Router(config-l2vpn-bg-bd-evi)# commit
```

Running Configuration

```
/* Configuration for root and leaf */

l2vpn
 bridge group bg1
  bridge-domain bd_1
   interface Bundle-Ether400.1
   !
   interface Bundle-Ether401.1001
   !
   interface Bundle-Ether4701.2001
   !
   etree
    leaf
   !
   evi 200
   !
  !
 !
```

EVPN Multiple Services per Ethernet Segment

Table 24: Feature History Table

Feature Name	Release Information	Feature Description
EVPN Multiple Services per Ethernet Segment on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The EVPN multiple services per Ethernet segment is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E

EVPN Multiple Services per Ethernet Segment	Release 24.2.11	You can configure EVPN to run multiple services on a single Ethernet Segment (ES), which enables the efficient use of network resources. While the services run on the same physical hardware resource, each service can be associated with a different EVPN instance and separated from each other. This allows traffic segregation, which enables users to employ their own traffic management configurations. This feature is supported only on routers with the Q200 and 88-LC1-36EH line cards.

When you configure multiple services on an Ethernet Segment (ES), the services use the same physical hardware resource. This provides traffic segregation, so that the users can manage the traffic configurations effectively.

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

- EVPN E-Line Xconnect service
- Native EVPN E-LAN services

Note These services are supported only on all-active multi-homing scenario.

Configure EVPN Multiple Services per Ethernet Segment

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

Configuration Example

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

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

```
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.3 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 3
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.4 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 4
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit


/* Configure EVPN E-Line xconnect service */

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

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

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

```
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64: 22022
Router (config-evpn-evi-bgp)# route-target export 64: 22022
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# advertise-mac
Router (config-evpn-evi)# commit
Router (config-evpn-evi)# exit
```

Running Configuration

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


/* Configure EVPN E-Line xconnect service */
interface Bundle-Ether22001.11 l2transport
  encapsulation dot1q 1 second-dot1q 11
  rewrite ingress tag pop 2 symmetric
 !
l2vpn
xconnect group xg22001
p2p evpn-vpws-mclag-22001
 interface Bundle-Ether22001.11
 neighbor evpn evi 22101 target 220101 source 220301
!
  !
/* Configure Native EVPN */
Evpn
 interface Bundle-Ether22001
  ethernet-segment   identifier type 0 ff.ff.ff.ff.ff.ff.ff.ff.ee
  bgp route-target 2200.0001.0001
  !
  evi 24001
   bgp
    route-target import 64:24001
    route-target export 64:24001
   !
   evi 21006
    bgp
      route-target 64:100006
   !
    evi 22101
```

```
   bgp
     route-target import 64:22101
     route-target export 64:22101
    !
  evi 22021
   bgp
     route-target import 64:22021
     route-target export 64:22021
   !
   advertise-mac
 !
 evi 22022
  bgp
   route-target import 64:22022
   route-target export 64:22022
  !
   advertise-mac
 !
```

Verification

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

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

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

Layer 2 Fast Reroute

Table 25: Feature History Table

Feature Name	Release Information	Feature Description
Layer 2 Fast Reroute	Release 24.3.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) Fast reroute minimizes traffic loss by quickly redirecting traffic to a backup path in the event of a link failure, ensuring fast convergence and maintaining the service continuity. This feature introduces the **convergence reroute** command.

The Layer 2 Fast Reroute (L2 FRR) feature allows a router to quickly redirect traffic through a backup path when a primary link fails. This feature minimizes traffic loss and ensures rapid convergence.

The L2 FRR feature protects traffic loss when a link failure occurs on the PE-CE connection. When a PE is not able to send traffic to the CE due to a local link failure, the traffic is redirected to a peer PE. The peer PE then sends the traffic to the CE.

L2 FRR for E-LAN service

Regardless of the state of the attachment circuit (AC) in a network, it is recommended to associate local hosts (MAC addresses) with a bridge port. When L2 FRR is enabled and an AC goes down, the MAC addresses are not flushed and remain associated with the L2 FRR-enabled AC.

L2 FRR in All-Active Multihoming Mode

In an All-Active redundancy mode, the AC-backup function is enabled by default for fast redirection of traffic using the All-Active peer's service label. The hosts (or MAC addresses) are permanently associated with the AC.

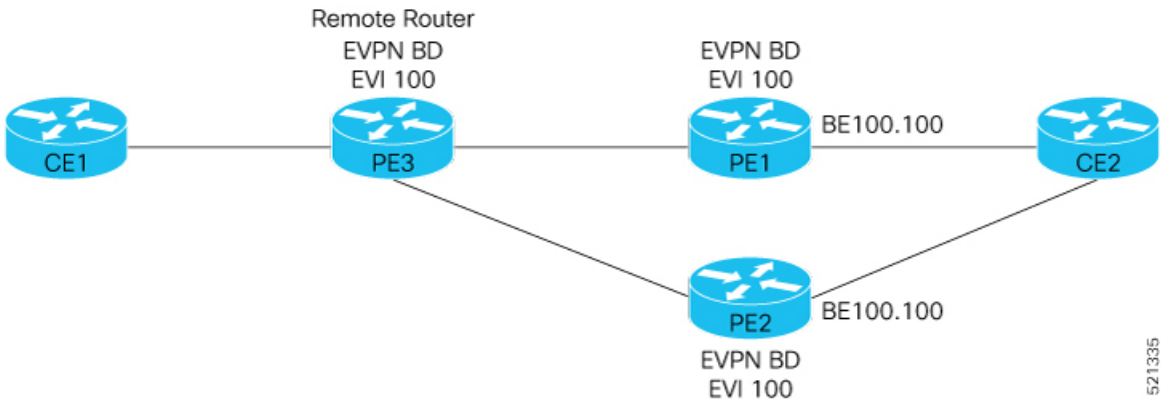
Benefits of L2 FRR

- Fast and predictable convergence.
- Fast failure notification even in large rings with a high number of nodes.
- Manual configuration for predictable failover behavior.
- No need to change the topology.

Topology

Consider a sample topology with EVPN multi-homing.

Figure 21: Layer 2 Fast Reroute



In this topology:

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

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

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

When the PE1-CE2 link goes down, L2 FRR is triggered on PE1. Traffic is redirected to PE2 until the convergence is complete.

- When you enable FRR on PE1, the logical backup path is pre-programmed in the hardware. When PE1 detects a failure on the access side (CE2), PE1 identifies the backup PE2 as has been programmed in the hardware.
- PE2 allocates and advertises an FRR label for protected traffic.
- All incoming traffic to PE1 is redirected to PE2 using this FRR label.
- PE1 encapsulates all the traffic with the FRR label of PE2 and forwards the traffic to PE2.
- PE2 receives the traffic with the FRR label and forwards the traffic to CE2.

Restrictions for Layer 2 Fast Reroute

This feature is supported on:

- BGP MPLS-Based EVPN E-LAN.
- PE devices in EVPN active-active or single-active mode.
- Unicast traffic.

This feature is not supported on:

- EVPN E-Line service.
- BUM traffic.

Configure Layer 2 Fast Reroute

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

Step 1 Associate the Ethernet segment with the bundle interface and enable L2 FRR using the **convergence reroute** command.

```
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether1
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 00.00.00.00.00.00.05.01.01
Router(config-evpn-ac-es)# convergence reroute
Router(config-evpn-ac-es)# commit
```

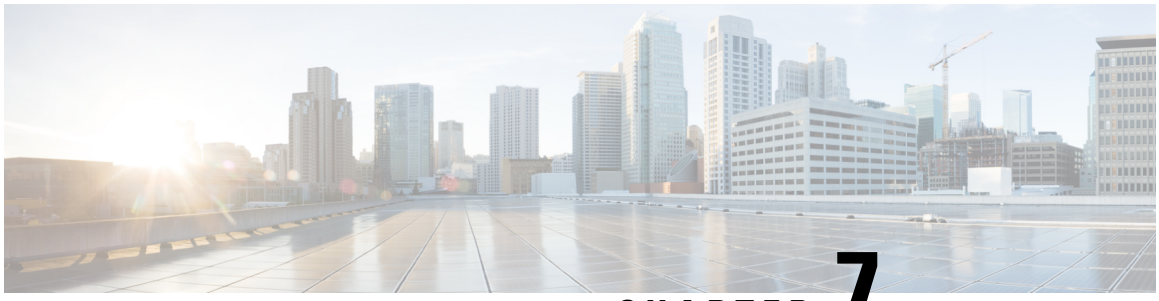
Step 2 View the running configuration.

```
evpn
 interface Bundle-Ether1
 ethernet-segment
   identifier type 0 00.00.00.00.00.00.05.01.01
   convergence reroute
```

Step 3 Verify the L2 FRR configuration.

```
Router# show evpn ethernet-segment interface bundle-Ether 1 carving detail
...
Ethernet Segment Id      Interface                         Nexthops
------------------------ --------------------------------- --------------------
0000.0000.0000.0005.0101 BE1                               N/A
  ES to BGP Gates   : B,M,R
  ES to L2FIB Gates : B,H
  Main port         :
     Interface name : Bundle-Ether1
     Interface MAC  : 0000.0000.0000
     IfHandle       : 0x00000000
     State          : Down
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0000.0000.0000.0005.0101
  ES Import RT      : 0000.0000.0000 (Incomplete Configuration)
  Topology          :
     Operational    : SH
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       : Reroute
  Peering Details   : 0 Nexthops
```

. . .



CHAPTER 7

CFM on EVPN

This chapter describes CFM on EVPN.

CFM on EVPN

Table 26: Feature History Table

Feature Name	Release Information	Feature Description
CFM on EVPN on 8212-48FH-M, 8711-32FH-M, 88-LC1-52Y8H-EM, and 88-LC1-12TH24FH-E	Release 24.3.1	The CFM on EVPN is now supported on these fixed systems and line cards: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
CFM on EVPN	Release 24.2.11	You can now proactively monitor connectivity and verify faults and isolate them for EVPN services. This is because Ethernet Connectivity Fault Management (CFM) is now available for EVPN and provides end-to-end service level OAM (Operations, Administration, and Maintenance) for EVPN services. This feature is supported only on routers with Q200 and 88-LC1-36EH line cards.

Ethernet Connectivity Fault Management (CFM) is an end-to-end per-service-instance Ethernet layer operation, administration, and management (OAM) protocol. It includes proactive connectivity monitoring, fault verification, and fault isolation for large networks.

For more information about Ethernet CFM, refer to the *Configuring Ethernet OAM* chapter in the *Interface and Hardware Component Configuration Guide for Cisco 8000 Series Routers*.

The CFM is now available for EVPN and provides end-to-end service level OAM (Operations, Administration, and Maintenance) for EVPN services. You can monitor the connections between the nodes using CFM in an EVPN network.

CFM is supported on EVPN network running with the following services.

- CFM on EVPN E-LAN Single-Homing: CFM on EVPN E-LAN is critical for service providers and enterprises to ensure high availability and reliability of their distributed Ethernet services.
- CFM on EVPN E-Line Single-Homing: CFM on EVPN E-Line ensures that service providers can monitor and maintain the health of point-to-point Ethernet services over a packet-switched network.

Restrictions for CFM on EVPN

- There is a possibility of unusual or unexpected results (also known as artifacts) appearing in the loopback or linktrace results. Either a loopback or linktrace may report multiple results for the same instance, or consecutive instances of a loopback and linktrace between the same two endpoints may produce different results.

Supported Offload Types and Timer Values

Continuity Check Messages (CCMs) are heartbeat messages exchanged periodically between all the Maintenance End Points (MEPs) in a service. Each MEP sends out multicast CCMs, and receives CCMs from all the other MEPs in the service. This allows each MEP to discover its peer MEPs, and to verify that there is connectivity between them. The offload type depends on where the CCMs are processed.

Only Non-offload is supported, where the CCMs are generated and processed by the CPU. The CCM timers for a CFM session on a or Bundle Interfaces are 1 second or greater.

CCM timers are the intervals in which CCMs are sent and received. If the CCMs are not received within the configured interval, the CFM MEP goes down. The following are the supported CCM timers for the Non-offload type:

- 1 sec
- 10 sec
- 1 min
- 10 min

CFM on EVPN E-LAN Single-Homing

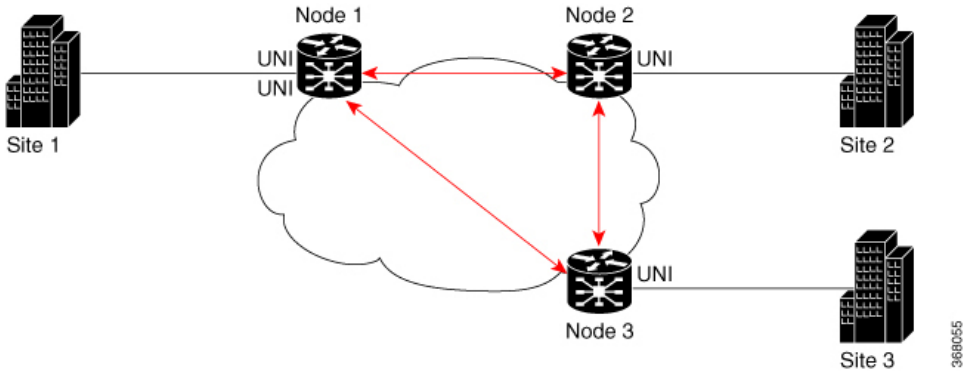
You can configure CFM on a network running with EVPN Emulated Local Area Network (E-LAN) services on single-homed devices to monitor the E-LAN services, thereby providing high-speed Layer 2 services with high resiliency.

Restrictions for CFM on EVPN E-LAN

- CFM on EVPN E-LAN is supported only on single-homed devices.

Configure CFM on EVPN E-LAN Single-Homing

Figure 22: CFM on EVPN E-LAN: Full Mesh Topology



Nodes 1, 2 and 3 in this topology are Cisco routers and are connected to each other.

Configuring CFM on EVPN E-LAN involves the following tasks:

- Enabling CFM service continuity check.
- Configuring Maintenance End Point (MEP) cross-check to validate the liveliness and consistency of remote MEPs in the network.
- Enabling CFM for the interface.

Configuration Example for CFM on EVPN E-LAN: Full Mesh Topology

```
/* Enable CFM continuity check */
Router# ethernet cfm
Router(config-cfm# domain bd-domain level 1 id null
Router(config-cfm-dmn)# service bd-domain bridge group bg-elan bridge-domain bd-elan id
icc-based MC MCMC
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-dmn-svc)# mep-id 1112
Router(config-cfm-dmn-svc)# mep-id 1113
Router(config-cfm-dmn-svc)# commit
```

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

```
/* Enable CFM on the interface */
Router(config)# interface TenGigE 0/0/0/2.100 l2transport
Router(config-subif)# description bg-elan
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# rewrite ingress tag pop 1 symmetric
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 1111
Router(config-if-cfm-mep)# commit
```

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

Running Configuration for CFM on EVPN E-LAN: Full Mesh Topology

This sections shows the running configuration on node 1.

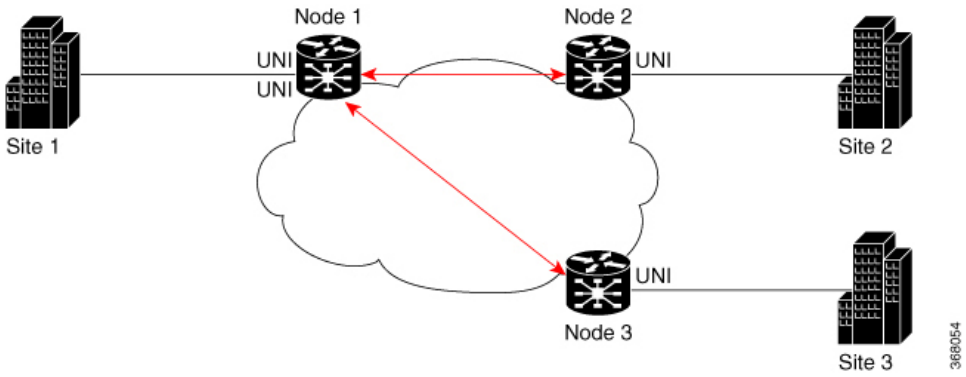
```
ethernet cfm
 domain bd-domain level 1 id null
  service bd-domain bridge group bg-elan bridge-domain bd-elan id icc-based MC MCMC
   continuity-check interval 10s
   mep crosscheck
    mep-id 1112
    mep-id 1113
   !
  !
 !
!

interface TenGigE 0/0/0/2.100 l2transport
 description bg-elan
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 1111
 !
```

Verification

The following outputs show the CFM configuration.

```
Router# show run ethernet cfm

ethernet cfm
  domain bd-domain level 1 id null
  service bd-domain bridge group bg-elan bridge-domain bd-elan id icc-based MC MCMC
   continuity-check interval 10s
   mep crosscheck
    mep-id 1112
    mep-id 1113

Router# show run interface TenGigE 0/0/0/2.100

interface TenGigE 0/0/0/2.100 l2transport
 description bg-elan
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 1111
```

Figure 23: CFM on EVPN E-LAN: Hub and Spoke Topology



Nodes 1, 2 and 3 in this topology are Cisco routers. Node 1 acts as the hub, whereas Node 2 and Node 3 are connected to the hub, Node1.

Configuration Example for CFM on EVPN E-LAN: Hub and Spoke Topology

The CFM configuration for the hub and spoke topology remains the same as that of full mesh topology mentioned above, except for these additional steps for SLA profile configuration to be done under the interface. You need to configure the SLA profile between the hub and the spokes to ensure continuous network services.

```
/* 1112 and 1113 in this example, are the mep-id values of node 2 and node 3 */
Router(config)#interface TenGigE 0/0/0/2.100 l2transport
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 1111
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1112
Router(config-if-cfm-mep)# sla operation profile test-profile2 target mep-id 1112
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1113
Router(config-if-cfm-mep)# sla operation profile test-profile2 target mep-id 1113
Router(config-if-cfm-mep)# commit
```

Running Configuration for CFM on EVPN E-LAN: Hub and Spoke Topology

This sections shows the running configuration on node 1.

```
interface TenGigE 0/0/0/2.100 l2transport
 description bg-elan
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 1111
   sla operation profile test-profile1 target mep-id 1112
   sla operation profile test-profile2 target mep-id 1112
   sla operation profile test-profile1 target mep-id 1113
   sla operation profile test-profile2 target mep-id 1113
 !
```

Configuration Examples for Different Types of Domains

Example 1: Configuration of Up MEPs with same domain and level on the following:

- Multiple AC per bridge domain on local

- Single AC per bridge domain on remote

```
/* Enable Ethernet CFM continuity check*/
Router(config)# ethernet cfm
Router(config-cfm)# domain BD-DOMAIN level 4 id null
Router(config-cfm-dmn)# service BD-SERVICE bridge group ELAN_FUNC_3 $
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 5
Router(config-cfm-xcheck)# mep-id 1101
Router(config-cfm-xcheck)# mep-id 1103

/* Configure bridge domain and assign interfaces to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group ELAN_FUNC_3
Router(config-l2vpn-bg)# bridge-domain FUNC_3
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/0.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/1.2
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 101

/* Enable CFM on the interfaces */
Router(config)# interface TenGigE0/0/0/0.1 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain BD-DOMAIN service BD-SERVICE mep-id 1103


Router(config)# interface TenGigE0/0/0/1.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain BD-DOMAIN service BD-SERVICE mep-id 5
```

Running Configuration

```
ethernet cfm
 domain BD-DOMAIN level 4 id null
  service BD-SERVICE bridge group ELAN_FUNC_3 bridge-domain FUNC_3 id number 100
   continuity-check interval 10s
   mep crosscheck
    mep-id 5
    mep-id 1101
    mep-id 1103

l2vpn
 bridge group ELAN_FUNC_3
  bridge-domain FUNC_3
   interface TenGigE0/0/0/0.1
   !
   interface TenGigE0/0/0/1.2
   !
   evi 101

interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 1103
```

```
interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 5
```

Example 2: Configuration of multiple Up MEPs on AC interfaces that are part of the same bridge domain:

```
/* Enable Ethernet CFM continuity check */
Router(config)# ethernet cfm
Router(config-cfm)# domain BD-DOMAIN level 4 id null
Router(config-cfm-dmn)# service BD-SERVICE bridge group ELAN_FUNC_3 bridge-domain FUNC_3 $
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 1
Router(config-cfm-xcheck)# mep-id 2
Router(config-cfm-xcheck)# mep-id 21
Router(config-cfm-xcheck)# mep-id 22

/* Enable Ethernet CFM continuity check for another domain level */
Router(config)# ethernet cfm
Router(config-cfm)# domain BD-DOMAIN1 level 3 id null
Router(config-cfm-dmn)# service BD-SERVICE1 bridge group ELAN_FUNC_3 bridge-domain FUNC_3
$
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 1001
Router(config-cfm-xcheck)# mep-id 1021
Router(config-cfm-xcheck)# mep-id 2001
Router(config-cfm-xcheck)# mep-id 2021

/* Configure bridge domain and assign interfaces to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group ELAN_FUNC_3
Router(config-l2vpn-bg)# bridge-domain FUNC_3
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/0
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 101

/* Enable CFM on the interfaces with multiple MEPs */
Router(config)# interface TenGigE0/0/0/0.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain BD-DOMAIN service BD-SERVICE mep-id 21
Router(config-if-cfm-mep)# exit
Router(config-if-cfm)# mep domain BD-DOMAIN1 service BD-SERVICE1 mep-id 1021

Router(config)# interface TenGigE0/0/0/1.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain BD-DOMAIN service BD-SERVICE mep-id 22
Router(config-if-cfm-mep)# exit
Router(config-if-cfm)# mep domain BD-DOMAIN1 service BD-SERVICE1 mep-id 2021
```

Running Configuration

```
ethernet cfm
 domain BD-DOMAIN level 4 id null
    service BD-SERVICE bridge group ELAN_FUNC_3 bridge-domain FUNC_3 id number 100
   continuity-check interval 10s
   mep crosscheck
    mep-id 1
    mep-id 2
    mep-id 21
    mep-id 22

 domain BD-DOMAIN1 level 3 id null
  service BD-SERVICE1 bridge group ELAN_FUNC_3 bridge-domain FUNC_3
   continuity-check interval 10s
   mep crosscheck
    mep-id 1001
    mep-id 1021
    mep-id 2001
    mep-id 2021


l2vpn
 bridge group ELAN_FUNC_3
  bridge-domain FUNC_3
   interface TenGigE0/0/0/0
   interface TenGigE0/0/0/1
   Interface TenGigE0/0/0/2
   evi 101

 interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 21
  mep domain BD-DOMAIN1 service BD-SERVICE1 mep-id 1021


interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain BD-DOMAIN service BD-SERVICE mep-id 22
  mep domain BD-DOMAIN1 service BD-SERVICE1 mep-id 2021
```

Example 3: Configuration of multiple services for different EVPN bridge domains:

```
/* Enable Ethernet CFM continuity check */
Router(config)# ethernet cfm
Router(config-cfm)#  domain evpn-bd level 4 id null
Router(config-cfm-dmn)# service evpn-bd1 bridge group BG1 bridge-domain BD1 $
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 5
Router(config-cfm-xcheck)# mep-id 6
Router(config-cfm-xcheck)# mep-id 1101
Router(config-cfm-xcheck)# mep-id 1103

/* Enable Ethernet CFM continuity check for another service */
Router(config)# ethernet cfm
Router(config-cfm)#  domain evpn-bd level 4 id null
```

```
Router(config-cfm-dmn)# service evpn-bd2 bridge group BG2 bridge-domain BD2 $
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 11
Router(config-cfm-xcheck)# mep-id 21
Router(config-cfm-xcheck)# mep-id 101

/* Configure bridge domain and assign interfaces to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group BG1
Router(config-l2vpn-bg)# bridge-domain BD1
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/0.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/1.2
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 101

/* Configure another bridge domain and assign interfaces to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group BG2
Router(config-l2vpn-bg)# bridge-domain BD2
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/2.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/5.2
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 201

/* Enable CFM on the interfaces with different services*/
Router(config)# interface TenGigE0/0/0/0.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd service evpn-bd1 mep-id 1103

Router(config)# interface TenGigE0/0/0/1.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd service evpn-bd1 mep-id 5

Router(config)# interface TenGigE0/0/0/2.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd service evpn-bd2 mep-id 101

Router(config)# interface TenGigE0/0/0/5.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)#   mep domain evpn-bd service evpn-bd2 mep-id 11
```

Running Configuration

```
ethernet cfm
 domain evpn-bd level 4 id null
  service evpn-bd1 bridge group BG1 bridge-domain BD1
   continuity-check interval 10s
   mep crosscheck
    mep-id 5
    mep-id 6
    mep-id 1101
    mep-id 1103
  service evpn-bd2 bridge group BG2 bridge-domain BD2
```

```
   continuity-check interval 10s
   mep crosscheck
    mep-id 11
    mep-id 21
    mep-id 101

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

interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 1103

interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 5

interface TenGigE0/0/0/2.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd service evpn-bd2 mep-id 101

interface TenGigE0/0/0/5.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd service evpn-bd2 mep-id 11
```

Example 4: Configuration of different EVPN bridge domains on different domain levels:

```
/* Enable Ethernet CFM continuity check */
Router(config)# ethernet cfm
Router(config-cfm)# domain evpn-bd level 4 id null
Router(config-cfm-dmn)# service evpn-bd1 bridge group BG1 bridge-domain BD1 $
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 5
Router(config-cfm-xcheck)# mep-id 6
Router(config-cfm-xcheck)# mep-id 1101
Router(config-cfm-xcheck)# mep-id 1103

/* Enable Ethernet CFM continuity check for another domain and service */
Router(config)# ethernet cfm
Router(config-cfm)# domain evpn-bd2 level 3 id null
Router(config-cfm-dmn)# service evpn-bd2 bridge group BG2 bridge-domain BD2 $
Router(config-cfm-dmn-svc)# continuity-check interval 10s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-xcheck)# mep-id 11
Router(config-cfm-xcheck)# mep-id 21
```

```
Router(config-cfm-xcheck)# mep-id 101
Router(config-cfm-xcheck)# mep-id 201

/* Configure bridge domain and assign interfaces to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group BG1
Router(config-l2vpn-bg)# bridge-domain BD1
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/0.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/1.2
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 101

/* Configure another bridge domain and assign interfaces to the bridge domain */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group BG2
Router(config-l2vpn-bg)# bridge-domain BD2
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/2.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface TenGigE0/0/0/5.2
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# evi 201

/* Enable CFM on the interfaces with different domains and services*/
Router(config)# interface TenGigE0/0/0/0.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd service evpn-bd1 mep-id 1103

Router(config)# interface TenGigE0/0/0/1.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd service evpn-bd1 mep-id 5

Router(config)# interface TenGigE0/0/0/2.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd2 service evpn-bd2 mep-id 101

Router(config)# interface TenGigE0/0/0/5.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain evpn-bd2 service evpn-bd2 mep-id 11
```

Running Configuration

```
ethernet cfm
 domain evpn-bd level 4 id null
  service evpn-bd1 bridge group BG1 bridge-domain BD1
   continuity-check interval 10s
   mep crosscheck
    mep-id 5
    mep-id 6
    mep-id 1101
    mep-id 1103
   !
  !
 !
 domain evpn-bd2 level 3 id null
  service evpn-bd2 bridge group BG2 bridge-domain BD2
   continuity-check interval 10s
```

```
   mep crosscheck
    mep-id 11
    mep-id 21
    mep-id 101
    mep-id 201
   !
  !
 !
!
l2vpn
 bridge group BG1
  bridge-domain BD1
   interface TenGigE0/0/0/0.1
   !
   interface TenGigE0/0/0/1.2
   !
   evi 101
   !
  !
 !
 bridge group BG2
  bridge-domain BD2
   interface TenGigE0/0/0/2.1
   !
   interface TenGigE0/0/0/5.2
   !
   evi 201
   !
  !
 !
!
interface TenGigE0/0/0/0.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 1103
  !
 !
!

interface TenGigE0/0/0/1.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd service evpn-bd1 mep-id 5
  !
 !
!

interface TenGigE0/0/0/2.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain evpn-bd2 service evpn-bd2 mep-id 101
  !
 !
!

interface TenGigE0/0/0/5.2 l2transport
 encapsulation dot1q 2
 ethernet cfm
  mep domain evpn-bd2 service evpn-bd2 mep-id 11
  !
 !
!
```

CFM on EVPN E-Line Single-Homing

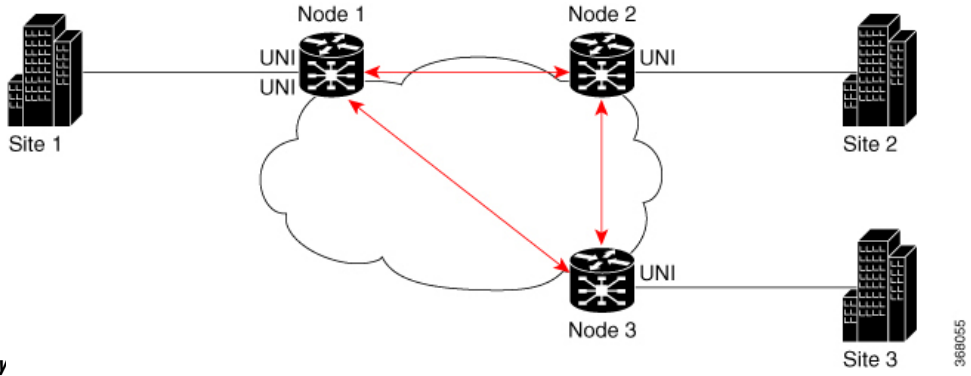
You can configure CFM on a network running with EVPN E-Line single-homed devices to monitor the E-Line services, that provides high-speed Layer 2 services with high resiliency.

Restrictions for CFM on EVPN E-Line

- CFM up MEP is supported only on single-homing Layer 2 main and subinterfaces.

Configure CFM on EVPN E-Line Single-Homing

Figure 24: CFM on EVPN E-Line: Full Mesh



Topology

Nodes 1, 2 and 3 in this topology are Cisco routers and are connected to each other.

Configuring CFM on EVPN E-Line involves the following tasks:

- Enabling CFM service continuity check.
- Configuring Maintenance End Point (MEP) cross-check to validate the liveliness and consistency of remote MEPs in the network.
- Enabling CFM for the interface.

Configuration Example

```
/* Enable CFM continuity check */
Router# ethernet cfm
Router(config-cfm# domain xcup1 level 7 id null
Router(config-cfm-dmn)# service xcup1 xconnect group evpn_vpws_Bundle_ether203 p2p
evpn_vpws-100 id number 4001
Router(config-cfm-dmn-svc)# continuity-check interval 1s

/* Configure MEP cross-check */
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-dmn-svc)# mep-id 4001
Router(config-cfm-dmn-svc)# commit
```

Repeat the above configurations for node 2 and node 3, with the respective mep-id values. For node 2, configure MEP cross-check with respective mep-id values of node 1 and node 3 (2001 and 3001 respectively, in this

example). For node 3, configure MEP cross-check with respective mep-id values of node 1 and node 2 (4001 and 2001 respectively, in this example).

```
/* Enable CFM on the interface */
Router# configure
Router(config)# interface Bundle-Ether203.2001 l2transport
Router(config-subif)# encapsulation dot1q 2001
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain xcup1 service xcup1 mep-id 2001
Router(config-if-cfm-mep)# commit
```

You must repeat the above configurations for node 2 and node 3, with the respective *mep-id* values.

Running Configuration

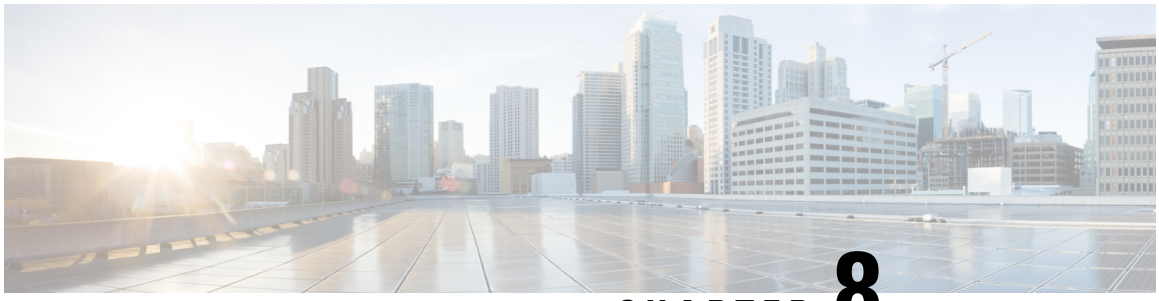
This sections shows the running configuration on node 1.

```
ethernet cfm
 domain xcup1 level 7 id null
  service xcup1 xconnect group evpn_vpws_Bundle_ether203 p2p evpn_vpws-100 id number 4001
   continuity-check interval 1s
   mep crosscheck
    mep-id 4001
   !
  !
 !
!
interface Bundle-Ether203.2001 l2transport
 encapsulation dot1q 2001
 ethernet cfm
  mep domain xcup1 service xcup1 mep-id 2001
 !
```

Verification

The following example shows CFM configuration.

```
Router# show ethernet cfm services

   Summary for Domain xcup1 (level 7), Service xcup1
================================================================================
Domain MIB index: 1, Service MIB index: 1
Domain ID: NULL, Service ID: UINT: 4001
Service configured on P2P cross-connect evpn_vpws-100 in group evpn_vpws_Bundle_ether
CCM interval: 1s
Local MEPs: 1 total
Peer MEPs: 2 total
MIPs: 0, MIP creation rule: always
```



CHAPTER 8

EVPN MPLS Transport

EVPN Bridging and E-Line Services over BGP-LU Underlay

Table 27: Feature History Table

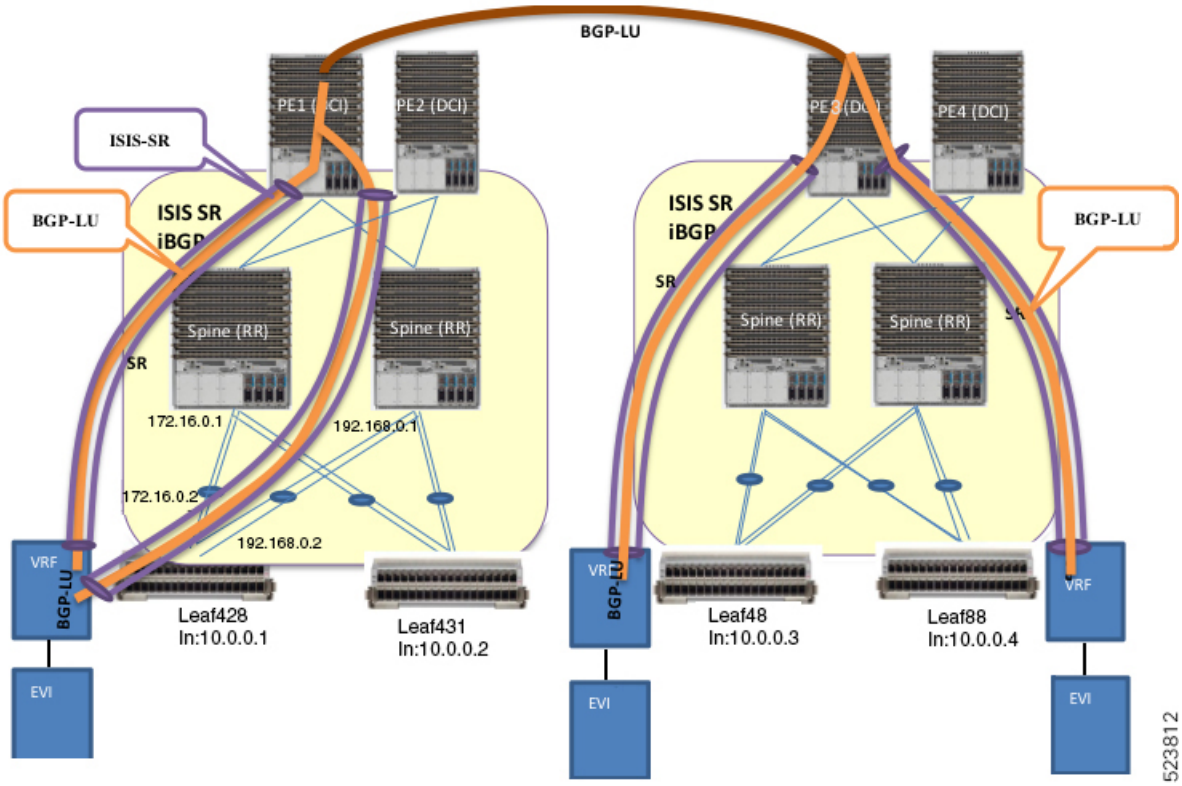
Feature Name	Release Information	Feature Description
EVPN Bridging and E-Line Services over BGP-LU Underlay	Release 7.11.1	You can configure end-to-end services between data centers using the BGP Labeled Unicast (BGP-LU) underlay. This feature allows you to configure various EVPN and E-Line services, and enables load balancing at transport, BGP-LU, and service level.

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

This feature supports the following services:

- EVPN Aliasing over BGP-LU using IGP (SR or non-SR (LDP or IGP))
- E-Line over BGP-LU using IGP

Figure 25: EVPN Bridging and E-Line Services over BGP-LU Underlay



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

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

  For example, at Leaf428, BGP-LU routes are learnt for remote loopback 10.0.0.3 and 10.0.0.4.

Restrictions for EVPN Bridging and E-Line Services over BGP-LU Underlay

The following EVPN services are not supported over BGP-LU over IGP with L2 unicast and BUM traffic:

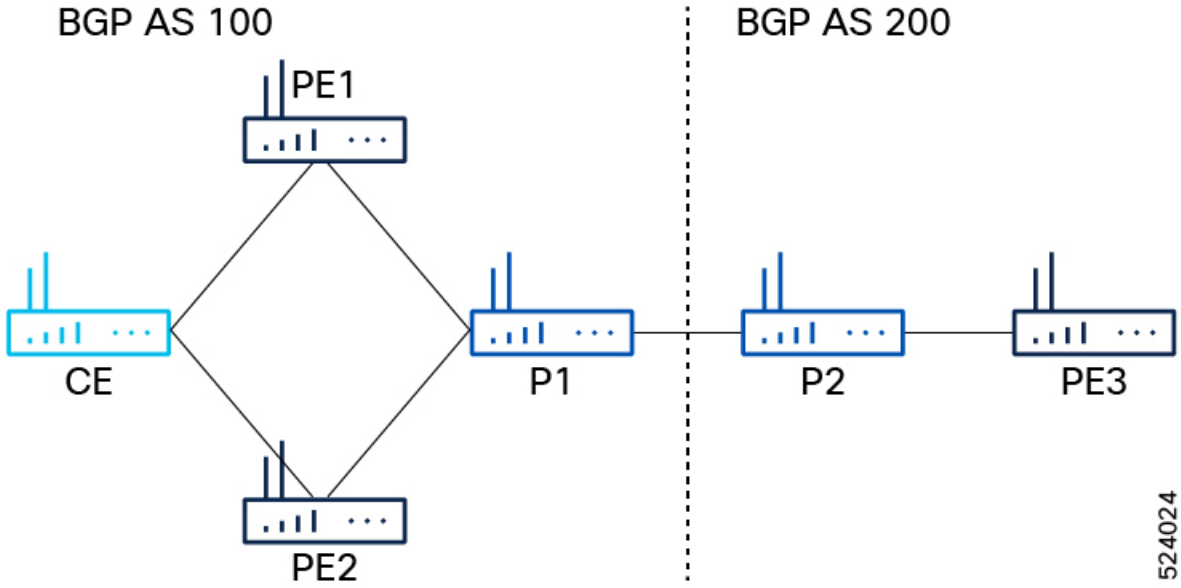
- EVPN E-LAN and E-LINE for EVPN single-active multi-homing mode.

- EVPN IRB with VRF (intra-subnet).
- Load Balancing for EVPN Bridging over BGP-LU with Multipaths.

Configure EVPN Bridging and E-Line Services over BGP-LU Underlay

Perform these tasks to configure the EVPN Bridging and E-Line Services over BGP-LU Underlay feature. Consider an example where the routers are connected as follows:

Figure 26:



- The topology consists of :
  - P routers: P1 and P2.
  - PE routers: PE1, PE2, PE3
- P1 is connected to P2.
- P1 is connected to PE1 and PE2 with BGP AS 100.
- P2 is connected to PE3 with BGP AS 200.

Configuration Example

Configure IGP, MPLS, and BGP on PE1 and PE2. The configuration is similar on both the routers.

```
/* Configure IGP */
IGP configuration is a pre-requisite to configure EVPN. IGP can be OSPF or ISIS.
Router(config)#router ospf pyats_test
Router(config-ospf)#router-id 54.54.54.54
Router(config-ospf)#redistribute bgp 100
Router(config-ospf)#mpls ldp sync
Router(config-ospf)#mpls ldp auto-config
```

```
Router(config-ospf)#area 0
Router(config-ospf-ar)#interface loopback0
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface FourHundredGigE0/0/0/2
Router(config-ospf-ar-if)#commit

/* Configure MPLS */
Router(config)# mpls ldp
Router(config-ldp)# router-id 54.54.54.54
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# label
Router(config-ldp-af-lbl)# local
Router(config-ldp-af-lbl)# allocate for host-routes
Router(config-ldp-af-lbl-lcl)# root
Router(config)# mpls ldp
Router(config-ldp)# interface FourHundredGigE0/0/0/2

/* Configure BGP */
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 54.54.54.54
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#exit
Router(config-bgp)#address-family l2vpn evpn
Router(config-bgp-af)#retain route-target all
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor-group IBGP-PEERS
Router(config-bgp-nbrgrp)#remote-as 100
Router(config-bgp-nbrgrp)#update-source loopback0
Router(config-bgp-nbrgrp)#address-family ipv4 unicast
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family l2vpn evpn

/* Configure iBGP peer on P1 */
Router(config-bgp)# neighbor 52.52.52.52
Router(config-bgp-nbr)# use neighbor-group IBGP-PEERS

/* Configure iBGP peer on PE2 */
Router(config-bgp)# neighbor 55.55.55.55
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# use neighbor-group IBGP-PEERS
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family l2vpn evpn
```

Configure IGP, MPLS, and BGP on PE3.

```
/* Configure IGP */
Router# configure
Router(config)#router ospf pyats_test
Router(config-ospf)#router-id 51.51.51.51
Router(config-ospf)#redistribute bgp 200
Router(config-ospf)#mpls ldp sync
Router(config-ospf)#mpls ldp auto-config
Router(config-ospf)#area 0
Router(config-ospf-ar)#interface loopback0
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface FourHundredGigE0/0/0/10
Router(config-ospf-ar-if)#commit

/* Configure MPLS */
Router(config)# mpls ldp
Router(config-ldp)# router-id 51.51.51.51
Router(config-ldp)# address-family ipv4
```

```
Router(config-ldp-af)# label
Router(config-ldp-af-lbl)# local
Router(config-ldp-af-lbl-lcl)# allocate for host-routes
Router(config-ldp-af-lbl-lcl)# root
Router(config)# mpls ldp
Router(config-ldp)# interface FourHundredGigE0/0/0/10

/* Configure BGP */
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 54.54.54.54
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#exit
Router(config-bgp)#address-family l2vpn evpn
Router(config-bgp-af)#retain route-target all
Router(config-bgp-af)#exit
Router(config-bgp)# neighbor 56.56.56.56
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#address-family l2vpn evpn

/* Configure P2 as route reflector */
Router(config)# prefix-set LOOPBACKS
Router(config-pfx)# 53.53.53.53,
Router(config-pfx)# 54.54.54.54,
Router(config-pfx)# 55.55.55.55,
Router(config-pfx)# 52.52.52.52,
Router(config-pfx)# 56.56.56.56,
Router(config-pfx)# 51.51.51.51
Router(config-pfx)# end-set

Router(config)# route-policy passall
Router(config-rpl)# pass
Router(config-rpl)# end-policy


Router(config)# route-policy MATCH_LOOPBACKS
Router(config-rpl)#if destination in LOOPBACKS then
Router(config-rpl-if)#pass
Router(config-rpl-if)#else
Router(config-rpl-else)#drop
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#

/* Configure route policy, IGP, MPLS, and BGP on P2 */
Router(config)#router ospf pyats_test
Router(config-ospf)#router-id 56.56.56.56
Router(config-ospf)#redistribute bgp 200
Router(config-ospf)#mpls ldp sync
Router(config-ospf)#mpls ldp auto-config
Router(config-ospf)#area 0
Router(config-ospf-ar)#interface loopback0
Router(config-ospf-ar-if)#passive enable
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface FourHundredGigE0/0/0/2

Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 100.0.0.1/32 FourHundredGigE0/0/0/5

Router(config)# mpls ldp
Router(config-ldp)# router-id 56.56.56.56
```

```
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# label
Router(config-ldp-af-lbl)# local
Router(config-ldp-af-lbl-lcl)# allocate for host-routes
Router(config-ldp-af-lbl-lcl)# root
Router(config)# mpls ldp
Router(config-ldp)# interface FourHundredGigE0/0/0/4

/* Configure router reflector client, which is essential for copying the EVPN routes between
 the AS */
Router(config)#router bgp 200
Router(config-bgp)#bgp router-id 56.56.56.56
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#network 51.51.51.51/32
Router(config-bgp-af)#network 52.52.52.52/32
Router(config-bgp-af)#network 53.53.53.53/32
Router(config-bgp-af)#network 54.54.54.54/32
Router(config-bgp-af)#network 55.55.55.55/32
Router(config-bgp-af)#network 56.56.56.56/32
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#redistribute ospf 0
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#address-family l2vpn evpn
Router(config-bgp-af)#retain route-target all
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor-group IBGP-PEERS
Router(config-bgp-nbrgrp)#remote-as 200
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#address-family l2vpn evpn
Router(config-bgp-nbr-af)#route-reflector-client

/* Configure P1 as eBGP neighbor */
Router(config-bgp)# neighbor 100.0.0.1
Router(config-bgp-nbr)#remote-as 100
Router(config-bgp-nbr)#ebgp-multihop 255
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#next-hop-self
Router(config-bgp-nbr-af)#route-policy passall in
Router(config-bgp-nbr-af)#route-policy MATCH_LOOPBACKS out
Router(config-bgp-nbr-af)#send-extended-community-ebgp
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#address-family l2vpn evpn
Router(config-bgp-nbr-af)#route-policy passall in
Router(config-bgp-nbr-af)#route-policy passall out
Router(config-bgp-nbr-af)#next-hop-unchanged

/* Configure PE3 as iBGP neighbor */
Router(config-bgp)#neighbor 51.51.51.51
Router(config-bgp-nbr)#use neighbor-group IBGP-PEERS
```

For P1, the iBGP peers are PE1 and PE2, and the eBGP peer is P2.

```
/* Configure P1 as route reflector */
Router(config)# prefix-set LOOPBACKS
Router(config-pfx)# 53.53.53.53,
Router(config-pfx)# 54.54.54.54,
Router(config-pfx)# 55.55.55.55,
Router(config-pfx)# 52.52.52.52,
Router(config-pfx)# 56.56.56.56,
```

```
Router(config-pfx)# 51.51.51.51
Router(config-pfx)# end-set

Router(config)# route-policy passall
Router(config-rpl)# pass
Router(config-rpl)# end-policy


Router(config)# route-policy MATCH_LOOPBACKS
Router(config-rpl)#if destination in LOOPBACKS then
Router(config-rpl-if)#pass
Router(config-rpl-if)#else
Router(config-rpl-else)#drop
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#

/* Configure route policy, IGP, MPLS, and BGP on P1 */
Router(config)#router ospf pyats_test
Router(config-ospf)#router-id 52.52.52.52
Router(config-ospf)#redistribute bgp 100
Router(config-ospf)#mpls ldp sync
Router(config-ospf)#mpls ldp auto-config
Router(config-ospf)#area 0
Router(config-ospf-ar)#interface loopback0
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface FourHundredGigE0/0/0/11
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface FourHundredGigE0/0/0/12

Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 100.0.0.2/32 FourHundredGigE0/0/0/13

Router(config)# mpls ldp
Router(config-ldp)# router-id 52.52.52
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# label
Router(config-ldp-af-lbl)# local
Router(config-ldp-af-lbl-lcl)# allocate for host-routes
Router(config-ldp-af-lbl-lcl)# root
Router(config)# mpls ldp
Router(config-ldp)# interface FourHundredGigE0/0/0/11
Router(config-ldp-if)# exit
Router(config-ldp)# interface FourHundredGigE0/0/0/12


/* Configure router reflector client */
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 52.52.52.52
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#network 51.51.51.51/32
Router(config-bgp-af)#network 52.52.52.52/32
Router(config-bgp-af)#network 53.53.53.53/32
Router(config-bgp-af)#network 54.54.54.54/32
Router(config-bgp-af)#network 55.55.55.55/32
Router(config-bgp-af)#network 56.56.56.56/32
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#redistribute ospf 0
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#address-family l2vpn evpn
Router(config-bgp-af)#retain route-target all
Router(config-bgp-af)#exit
```

```
Router(config-bgp)#neighbor-group IBGP-PEERS
Router(config-bgp-nbrgrp)#remote-as 100
Router(config-bgp-nbr)#update-source loopback0
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#address-family l2vpn evpn
Router(config-bgp-nbr-af)#route-reflector-client

/* Configure P2 as eBGP neighbor */
Router(config-bgp)# neighbor 100.0.0.2
Router(config-bgp-nbr)#remote-as 200
Router(config-bgp-nbr)#ebgp-multihop 255
Router(config-bgp-nbr)#address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)#next-hop-self
Router(config-bgp-nbr-af)#route-policy passall in
Router(config-bgp-nbr-af)#route-policy MATCH_LOOPBACKS out
Router(config-bgp-nbr-af)#send-extended-community-ebgp
Router(config-bgp-nbr-af)#exit
Router(config-bgp-nbr)#address-family l2vpn evpn
Router(config-bgp-nbr-af)#route-policy passall in
Router(config-bgp-nbr-af)#route-policy passall out
Router(config-bgp-nbr-af)#next-hop-unchanged

/* Configure PE1 and PE2 as iBGP neighbors */
Router(config-bgp)#neighbor 54.54.54.54
Router(config-bgp-nbr)#use neighbor-group IBGP-PEERS
Router(config-bgp-nbr)#exit
Router(config-bgp)#neighbor 55.55.55.55
Router(config-bgp-nbr)#use neighbor-group IBGP-PEERS
```

Configure L2VPN and EVPN on PE1, PE2, and PE3.

```
/* PE1 Configuration */

/* Configure Bridge Domain and EVI */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether3.1
Router(config-l2vpn-bg-bd-ac)# evi 1
Router(config-l2vpn-bg-bd-ac)# root

Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg2
Router(config-l2vpn-bg)# bridge-domain bd2
Router(config-l2vpn-bg-bd)# interface Bundle-Ether3.2
Router(config-l2vpn-bg-bd-ac)# evi 2

/* Configure EVPN EVI */
Router(config)# evpn
Router(config-evpn)# evi 1
Router(config-evpn-evi)# advertise-mac
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 2
Router(config-evpn-evi)# advertise-mac
```

Running Configuration

Verification

To verify that you have configured EVPN Bridging and E-Line Services over BGP-LU Underlay, use the following show commands.

```
Router# show evpn internal-label vpn-id 1 detail

VPN-ID     Encap  Ethernet Segment Id         EtherTag     Label
---------- ------ -------------------------- ----------   --------
1          MPLS   0040.0000.0000.0000.0001    0            24010
   Multi-paths resolved: TRUE (Remote all-active)
   Multi-paths Internal label: 24010
    EAD/ES  (ID:0x0000000000000652)
                54.54.54.54                                0
                55.55.55.55                                0
    EAD/EVI (ID:0x0000000000000649)
                54.54.54.54                                24000
                55.55.55.55                                24000
   Summary pathlist (ID 0x000000000000064d):
 0x02000001 (P) 54.54.54.54                                24000
 0x02000002 (P) 55.55.55.55                                24000

Router# show bgp l2vpn evpn route-type inclusive-mcast

BGP router identifier 51.51.51.51, local AS number 200
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP table nexthop route policy:
BGP main routing table version 100
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 51.51.51.51:1 (default for vrf bd1)
Route Distinguisher Version: 94
*> [3][0][32][51.51.51.51]/80
                      0.0.0.0                                  0 i N
*>i[3][0][32][54.54.54.54]/80
                      54.54.54.54                   100        0 100 i N
*>i[3][0][32][55.55.55.55]/80
                      55.55.55.55                   100        0 100 i N
Route Distinguisher: 51.51.51.51:2 (default for vrf bd2)
Route Distinguisher Version: 100
*> [3][0][32][51.51.51.51]/80
                      0.0.0.0                                  0 i N
*>i[3][0][32][54.54.54.54]/80
                      54.54.54.54                   100        0 100 i N
*>i[3][0][32][55.55.55.55]/80
                      55.55.55.55                   100        0 100 i N
Route Distinguisher: 54.54.54.54:1
Route Distinguisher Version: 92
*>i[3][0][32][54.54.54.54]/80
                      54.54.54.54                   100        0 100 i N
Route Distinguisher: 54.54.54.54:2
Route Distinguisher Version: 99
*>i[3][0][32][54.54.54.54]/80
                      54.54.54.54                   100        0 100 i N
Route Distinguisher: 55.55.55.55:1
Route Distinguisher Version: 67
*>i[3][0][32][55.55.55.55]/80
                      55.55.55.55                   100        0 100 i N
Route Distinguisher: 55.55.55.55:2
Route Distinguisher Version: 96
*>i[3][0][32][55.55.55.55]/80
```

```
                      55.55.55.55                   100      0 100 i N

Processed 10 prefixes, 10 paths
```

```
Router# show l2vpn forwarding bridge-domain mac location 0/RP0/CPU0

 To Resynchronize MAC table from the Network Processors, use the command...
    l2vpn resynchronize forwarding mac-address-table location <r/s/i>

Mac Address    Type    Learned from/Filtered on    LC learned Resync Age/Last Change Mapped
 to
-------------- ------- -------------------------- ---------- ----------------------
--------------
0000.cccc.dddd dynamic FH0/0/0/0.2                N/A        12 Mar 13:17:36        N/A
 --> MAC 0000.cccc.dddd was locally learned from interface FH0/0/0/0.2
0000.aaaa.bbbb EVPN    BD id: 1                   N/A        N/A                    N/A
 --> MAC 0000.aaaa.bbbb was advertised from PE1/PE2
```

```
Router# show bgp l2vpn evpn route-type mac-advertisement

BGP router identifier 51.51.51.51, local AS number 200
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP table nexthop route policy:
BGP main routing table version 100
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 51.51.51.51:2 (default for vrf bd2)
Route Distinguisher Version: 100
*>i[2][0][48][0000.aaaa.bbbb][0]/104 -->
                      54.54.54.54                   100      0 100 i N
* i                   55.55.55.55                   100      0 100 i N
*> [2][0][48][0000.cccc.dddd][0]/104 -->
                      0.0.0.0                                0 i N
Route Distinguisher: 54.54.54.54:2
Route Distinguisher Version: 99
*>i[2][0][48][0000.aaaa.bbbb][0]/104
                      54.54.54.54                   100      0 100 i N
Route Distinguisher: 55.55.55.55:2
Route Distinguisher Version: 96
*>i[2][0][48][0000.aaaa.bbbb][0]/104
                      55.55.55.55                   100      0 100 i N
Processed 4 prefixes, 5 paths
```

L2VPN Services over Segment Routing Traffic Engineering Tunnel

Segment Routing (SR) is a flexible and scalable way of performing source routing. The source device selects a path and encodes it in the packet header as an ordered list of segments. Segments are identifiers for any type of instruction.

Segment routing for traffic engineering (SR-TE) takes place through a tunnel between a source and destination pair. SR-TE uses the concept of source routing, where the source calculates the path and encodes it in the packet header as a segment. In SR-TE preferred path, each segment is an end-to-end path from the source to the destination, and instructs the routers in the provider core network to follow the specified path instead of the shortest path calculated by the IGP. The destination is unaware of the presence of the tunnel.

The user can achieve better resilience and convergence for the network traffic, by transporting MPLS L2VPN services using segment routing, instead of MPLS LDP. Segment routing can be directly applied to the MPLS architecture without changing the forwarding plane. In a segment-routing network that uses the MPLS data plane, LDP or other signaling protocol is not required; instead label distribution is performed by IGP. Removing protocols from the network simplifies its operation and makes it more robust and stable by eliminating the need for protocol interaction. Segment routing utilizes the network bandwidth more effectively than traditional MPLS networks and offers lower latency.

Preferred tunnel path functionality allows you to map pseudowires to specific traffic-engineering tunnel paths. Attachment circuits are cross-connected to specific SR traffic engineering tunnel interfaces instead of remote PE router IP addresses reachable using IGP or LDP. Using preferred tunnel path, the traffic engineering tunnel transports traffic from the source to destination PE routers. A path is selected for an SR Policy when the path is valid and its preference is the best (highest value) among all the candidate paths of the SR Policy.

L2VPN Preferred path

L2VPN preferred-path configuration allows you to steer pseudowire (PW) traffic over an SR-TE tunnel at the granularity of per PW level. It is recommended to use preferred-path configuration together with statically configured SR policy. EVPN ODN configuration allows you to steer PW traffic over an SR-TE tunnel at the granularity of per EVPN instance (EVI) level. When both the preferred-path and EVPN ODN configurations are used on a PW, preferred-path configuration takes precedence.

VPWS over SR-TE Policy

Table 28: Feature History Table

Feature Name	Release Information	Feature Description
VPWS over SR-TE Policy	Release 7.7.1	Using the SR-TE policy, you can now set the preferred path between two endpoints for Virtual Private Wire Service (VPWS). This gives you the advantage of a reduced number of dedicated networks to provision IP and VPN services across SR-TE tunnels. The VPWS is a method to provide Ethernet-based point to point communication over MPLS or IP networks.

VPWS connects two locations via a pseudowire (PW). PW traffic is encapsulated within an MPLS label stack that is used to route the traffic.

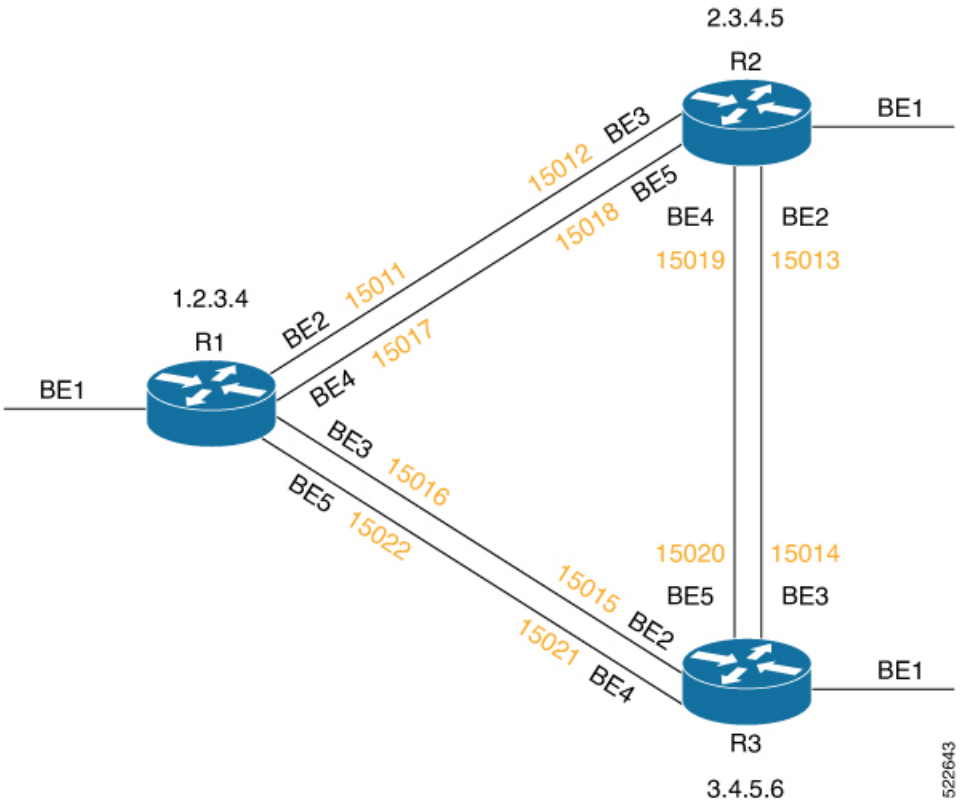
SR-TE uses SR policies to control the path the PW takes through the network, as opposed to using Label Distribution Protocol (LDP) to distribute labels.

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

Cisco 8000 series routers provide the following methods of implementing VPWS:

- EVPN VPWS
- LDP VPWS
- EVPN VPWS On-Demand Nexthop

Figure 27: VPWS over SR-TE



The image shows a PW between the routers R1 and R3. By default, the PW takes the direct path from R1 to R3.

You can configure an SR-TE policy with preferred path to steer the traffic to pass through R2.

VPWS PW over Preferred SR-TE using Statically Configured SR Policy

This feature allows you to set the preferred path between the two endpoints for EVPN VPWS PW or LDP VPWS PW using statically configured policy. This feature is supported on bundle attachment circuit (AC) and physical AC.

Restrictions

- EVPN VPWS SR policy is not supported on EVPN VPWS dual homing.
- EVPN validates if the route is for a single home nexthop, otherwise it issues an error message about a dangling SR TE policy, and continues to set up EVPN-VPWS without it. EVPN relies on 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 must manually configure the ESI value to zero to overwrite the auto-sense ESI as EVPN VPWS multihoming is not supported.

  To disable EVPN dual homing, configure bundle-Ether AC with ESI value set to zero.

```
evpn
interface Bundle-Ether12
  ethernet-segment
   identifier type 0 00.00.00.00.00.00.00.00.00
/* Or globally */
evpn
 ethernet-segment type 1 auto-generation-disable
```

Configure VPWS PW over Preferred SR-TE using Statically Configured SR Policy

Perform these tasks to ensure the successful configuration of VPWS PW over Preferred SR-TE using statically configured SR policy for EVPN VPWS or LDP VPWS:

- Configure SR in IGP (IS-IS)
- Configure Segment Routing
- Configure Segment List
- Configure Static SR-TE Policy
- Configure EVPN VPWS over Statically Configured Policy
- Configure LDP VPWS over Statically Configured Policy

Configure SR in IGP (IS-IS)

The following examples show how to enable SR in IS-IS and configure Adjacency-SID on BE2, BE3, BE4, and BE5.

```
/* Enable SR in IS-IS */
```

```
Router(config)# router isis isp
Router(config-isis)# net 01.0000.0000.0001.00
Router(config-isis)# distribute link-state instance-id 32 level 2 throttle 5
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# mpls traffic-eng level-1
Router(config-isis-af)# mpls traffic-eng router-id 1.2.3.4
Router(config-isis-af)# segment-routing mpls sr-prefer
Router(config-isis-af)# segment-routing prefix-sid-map advertise-local
Router(config-isis-af)# exit

/* Configure Adjacency-SID on BE2 */
Router(config-isis)# interface Bundle-Ether2
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# adjacency-sid index 11
Router(config-isis-if-af)# adjacency-sid absolute 15011
Router(config-isis-if-af)# exit

/* Configure Adjacency-SID on BE3 */
Router(config-isis)# interface Bundle-Ether3
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# adjacency-sid index 16
Router(config-isis-if-af)# adjacency-sid absolute 15016
Router(config-isis-if-af)# exit

/* Configure Adjacency-SID on BE4 */
Router(config-isis)# interface Bundle-Ether4
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# adjacency-sid index 17
Router(config-isis-if-af)# adjacency-sid absolute 15017
Router(config-isis-if-af)# exit

/* Configure Adjacency-SID on BE5 */
Router(config-isis)# interface Bundle-Ether5
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# adjacency-sid index 22
Router(config-isis-if-af)# adjacency-sid absolute 15022
Router(config-isis-if-af)# exit

/* Configure Prefix-SID */
Router(config-isis)# interface Loopback0
Router(config-isis-if)# passive
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# prefix-sid index 1001
```

Configure Segment Routing

The following examples show how to configure segment routing on BE2, BE3, BE4, and BE5.

```
Router(config)# segment-routing
Router(config-sr)# adjacency-sid
Router(config-sr)# exit

Router(config)# interface Bundle-Ether2
Router(config-if)# address-family ipv4 unicast
Router(config-if-af)# l2-adjacency-sid absolute 15011 next-hop 0.0.0.0
```

```
Router(config)# interface Bundle-Ether3
Router(config-if)# address-family ipv4 unicast
Router(config-if-af)# l2-adjacency-sid absolute 15016 next-hop 0.0.0.0

Router(config)# interface Bundle-Ether4
Router(config-if)# address-family ipv4 unicast
Router(config-if-af)# l2-adjacency-sid absolute 15017 next-hop 0.0.0.0

Router(config)# interface Bundle-Ether5
Router(config-if)# address-family ipv4 unicast
Router(config-if-af)# l2-adjacency-sid absolute 15022 next-hop 0.0.0.0
```

Configure Segment List

This example shows how to configure segment list for Adjacency-SID.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list segment_list_r3_1
Router(config-sr-te-sl)# index 10 mpls label 15011
Router(config-sr-te-sl)# index 20 mpls label 15013
Router(config-sr-te-sl)# exit

Router(config-sr-te)# segment-list segment_list_r3_2
Router(config-sr-te-sl)# index 10 mpls label 15011
Router(config-sr-te-sl)# index 20 mpls label 15019
Router(config-sr-te-sl)# exit

Router(config-sr-te)# segment-list segment_list_r3_3
Router(config-sr-te-sl)# index 10 mpls label 15017
Router(config-sr-te-sl)# index 20 mpls label 15013
Router(config-sr-te-sl)# exit

Router(config-sr-te)# segment-list segment_list_r3_4
Router(config-sr-te-sl)# index 10 mpls label 15017
Router(config-sr-te-sl)# index 20 mpls label 15019
Router(config-sr-te-sl)# exit
```

Configure Static SR-TE Policy

This example shows how to configure static SR-TE policy.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy policy_r3_1
Router(config-sr-te-policy)# color 10 end-point ipv4 3.4.5.6
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-pp-info)# explicit segment-list segment_list_r3_1
Router(config-sr-te-pp-info)# weight 10

Router(config-sr-te-pp-info)# explicit segment-list segment_list_r3_2
Router(config-sr-te-pp-info)# weight 10

Router(config-sr-te-pp-info)# explicit segment-list segment_list_r3_3
Router(config-sr-te-pp-info)# weight 10

Router(config-sr-te-pp-info)# explicit segment-list segment_list_r3_4
Router(config-sr-te-pp-info)# weight 10
```

Configure EVPN VPWS over Statically Configured Policy

This example shows how to configure EVPN VPWS over statically configured policy.

Note Use the auto-generated SR-TE policy name to attach the policy to the L2VPN instance. The auto-generated policy name is based on the policy color and end-point. Use the **show segment-routing traffic-eng policy candidate-path name** *policy_name* command to get the auto-generated policy name.

```
Router# show segment-routing traffic-eng policy candidate-path name policy_r3_1

SR-TE policy database
---------------------
Color: 10, End-point: 3.4.5.6
 Name: sr-te policy srte_c_10_ep_3.4.5.6
```

Set the preferred path with the SR-TE policy name.

```
Router(config)# l2vpn
Router(config-l2vpn)# pw-class c1
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# preferred-path sr-te policy srte_c_10_ep_3.4.5.6
!
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg
Router(config-l2vpn-xc)# p2p xc200
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether1.200
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 2 target 3 source 4
Router(config-l2vpn-xc-p2p-pw)# pw-class c1
!
```

Configure LDP VPWS over Statically Configured Policy

This example shows how to configure LDP VPWS over statically configured policy.

Note Use the auto-generated SR-TE policy name to attach the policy to the L2VPN instance. The auto-generated policy name is based on the policy color and end-point. Use the **show segment-routing traffic-eng policy candidate-path name** *policy_name* command to get the auto-generated policy name.

```
Router# show segment-routing traffic-eng policy candidate-path name policy_r3_1

SR-TE policy database
---------------------
Color: 10, End-point: 3.4.5.6
 Name: sr-te policy srte_c_10_ep_3.4.5.6
```

Set the preferred path with the SR-TE policy name.

```
Router(config)# l2vpn
Router(config-l2vpn)# pw-class c
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# preferred-path sr-te policy srte_c_10_ep_3.4.5.6
Router(config-l2vpn-pwc-mpls)# commit
Router(config-l2vpn-pwc-mpls)# exit

Router(config)# l2vpn
```

```
Router(config-l2vpn)# xconnect group xg
Router(config-l2vpn-xc)# p2p xc100
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether1.100
Router(config-l2vpn-xc-p2p)# neighbor ipv4 3.4.5.6 pw-id 100
Router(config-l2vpn-xc-p2p-pw)# pw-class c
```

Verify VPWS PW over Preferred SR-TE using Statically Configured SR Policy

The following show commands display the output for LDP VPWS PW configuration over preferred SR-TE. You can use the same show commands to view the EVPN VPWS PW configuration over preferred SR-TE.

```
Router# show l2vpn xconnect interface Bundle-Ether 1.100 detail
Mon May 16 14:19:42.833 UTC

Group xg, XC xc100, state is up; Interworking none
  AC: Bundle-Ether1.100, state is up
  PW: neighbor 3.4.5.6, PW ID 100, state is up ( established )
    PW class c, XC ID 0xa0000001
    Encapsulation MPLS, protocol LDP
    Source address 1.2.3.4
    PW type Ethernet, control word disabled, interworking none
    PW backup disable delay 0 sec
    Sequencing not set
   Preferred path Active : SR TE srte_c_10_ep_3.4.5.6 (BSID:24011, IFH:0xf000014), Statically
 configured, fallback enabled
    Ignore MTU mismatch: Disabled
    Transmit MTU zero: Disabled
    Tunnel : Up


Router# show segment-routing traffic-eng policy color 10 endpoint ipv4 3.4.5.6 detail
Mon May 16 14:02:16.550 UTC

SR-TE policy database
---------------------

Color: 10, End-point: 3.4.5.6
  Name: srte_c_10_ep_3.4.5.6
  Status:
    Admin: up  Operational: up for 00:05:09 (since May 16 13:57:06.627)


Router# show segment-routing traffic-eng forwarding policy color 10 endpoint ipv4 3.4.5.6
detail
Mon May 16 14:04:49.061 UTC

SR-TE Policy Forwarding database
--------------------------------

 Color: 10, End-point: 3.4.5.6
  Name: srte_c_10_ep_3.4.5.6
  Binding SID: 24011
  Active LSP:
    Candidate path:
      Preference: 100 (configuration)
      Name: policy_r3_1
    Local label: 24021
    Segment lists:
      SL[0]:
       Name: segment_list_r3_1
        Switched Packets/Bytes: 0/0
        Paths:
          Path[0]:
```

```
          Outgoing Label: 15013
          Outgoing Interfaces: Bundle-Ether2
.........

    SL[1]:
      Name: segment_list_r3_2
      Switched Packets/Bytes: 0/0
      Paths:
        Path[0]:
          Outgoing Label: 15019
          Outgoing Interfaces: Bundle-Ether2
.........

    SL[2]:
      Name: segment_list_r3_3
      Switched Packets/Bytes: 0/0
      Paths:
        Path[0]:
          Outgoing Label: 15013
          Outgoing Interfaces: Bundle-Ether4
.........

    SL[3]:
      Name: segment_list_r3_4
      Switched Packets/Bytes: 0/0
      Paths:
        Path[0]:
          Outgoing Label: 15019
          Outgoing Interfaces: Bundle-Ether4

.........
Policy Packets/Bytes Switched: 0/0
```

EVPN PW over Preferred SR-TE using On-Demand Nexthop SR Policy

This feature enables you to fetch the best path to send traffic from the source to destination in a point-to-point service using IOS XR Traffic Controller (XTC). On-Demand Nexthop (ODN) with SR-TE is supported on EVPN E-LAN and EVPN E-Line services.

When redistributing routing information across domains, provisioning of multi-domain services (Layer2 VPN and Layer 3 VPN) poses complexity and scalability issues. ODN with SR-TE feature delegates computation of an end-to-end Label Switched Path (LSP) to a path computation element (PCE). This PCE includes constraints and policies without any redistribution. It then installs the reapplied multi-domain LSP for the duration of the service into the local forwarding information base(FIB).

ODN uses BGP dynamic SR-TE capabilities and adds the path to the PCE. The PCE has the ability to find and download the end-to-end path based on the requirements. ODN triggers an SR-TE auto-tunnel based on the defined BGP policy. The PCE learns real-time topologies through BGP and/or IGP.

IOS XR Traffic Controller (XTC)

The path computation element (PCE) describes a set of procedures by which a path computation client (PCC) reports and delegates control of head-end tunnels sourced from the PCC to a PCE peer. The PCE peer requests the PCC to update and modify parameters of LSPs it controls. It also enables a PCC to allow the PCE to initiate computations and to perform network-wide orchestration.

Restrictions

- Maximum number of auto-provisioned SR-TE policies is 1000.

Configure EVPN VPWS PW over Preferred SR-TE using On-Demand Nexthop SR Policy

Perform the following steps to configure EVPN VPWS PW over preferred SR-TE using On-Demand Nexthop SR policy.

- Configure SR in IS-IS
- Configure SR-TE Policy
- Configure PCE and PCC
- Configure BGP
- Configure SR Color
- Configure EVPN Route Policy
- Configure EVPN VPWS over SR-TE Policy

Configure SR in IS-IS

The following examples show how to enable SR in IS-IS and configure interfaces BE2, BE3, BE4, and BE5.

```
/* Enable SR in IS-IS */
Router(config)# router isis isp
Router(config-isis)# net 01.0000.0000.0001.00
Router(config-isis)# distribute instance-id 32 level 2 throttle 5
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# mpls traffic-eng level-1
Router(config-isis-af)# mpls traffic-eng router-id 1.2.3.4
Router(config-isis-af)# segment-routing mpls sr-prefer
Router(config-isis-af)# segment-routing prefix-sid-map advertise-local
Router(config-isis-af)# exit

/* Configure interface BE2 */
Router(config-isis)# interface Bundle-Ether2
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit

/* Configure interface BE3 */
Router(config-isis)# interface Bundle-Ether3
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit
Router(config-isis-if)# exit

/* Configure interface BE4 */
Router(config-isis)# interface Bundle-Ether4
```

```
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit


/* Configure interface BE5 */
Router(config-isis)# interface Bundle-Ether5
Router(config-isis-if)# point-to-point
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# exit


/* Configure Loopback interface */
Router(config-isis)# interface Loopback0
Router(config-isis-if)# passive
Router(config-isis-if)# address-family ipv4 unicast
```

Configure SR-TE Policy

Configure SR-TE policy.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# on-demand color 1
Router(config-sr-te-color)# dynamic
Router(config-sr-te-color-dyn)# pcep
Router(config-sr-te-color-dyn-pce)# exit
Router(config-sr-te-color-dyn)# metric type igp
Router(config-sr-te-color-dyn)# exit
Router(config-sr-te-color)# exit

Router(config-sr-te)# on-demand color 2
Router(config-sr-te-color)# dynamic
Router(config-sr-te-color-dyn)# pcep
Router(config-sr-te-color-dyn-pce)# exit
Router(config-sr-te-color-dyn)# metric type igp
Router(config-sr-te-color-dyn)# exit
Router(config-sr-te-color)# exit
```

Configure PCE and PCC

Configure PCE and PCC on the routers.

```
/* Configure PCC on R1 */


Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# pcc
Router(config-sr-te-pcc)# source-address ipv4 1.2.3.4 >>>>> This is the node address
Router(config-sr-te-pcc)# pce address ipv4 3.4.5.6 >>>>> This is the PCE server address
Router(config-sr-te-pcc)# commit

/* Configure PCE on R3 */


Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# exit
Router(config)# pce
Router(config-pce)# address ipv4 3.4.5.6 >>>>> Configure the address only on a PCE server
Router(config-pce)# commit
```

Configure BGP

Configure BGP on the routers to establish neighborship with EVPN. When you enable an SR policy on R1, use the **nexthop validation color-extcomm sr-policy** command to instruct BGP that, instead of nexthop reachability validation of BGP routes, the validation is done for SR policy-installed color nexthop addresses. When the nexthop address of such a route is reachable, the route is added to the routing table.

```
Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 1.2.3.4
Router(config-bgp)# nexthop validation color-extcomm sr-policy
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp-af)# exit
!
Router(config-bgp)# neighbor 2.3.4.5
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp-af)# exit
!
Router(config-bgp)# neighbor 3.4.5.6
Router(config-bgp-nbr)# remote-as 100
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp-af)# exit
!
```

Configure SR Color

Configure SR colors on the routers.

```
Router(config)# extcommunity-set opaque color1
Router(config-ext)# 1
Router(config-ext)# end-set
!
Router(config)# extcommunity-set opaque color2
Router(config-ext)# 2
Router(config-ext)# end-set
!
```

Configure EVPN Route Policy

Configure EVPN route policy. This example shows how to define the route policy language and track the EVPN route.

```
Router(config)# route-policy route_policy_1
Router(config-rpl)# if evpn-route-type is 1 then
Router(config-rpl-if)# set extcommunity color color1
Router(config-rpl-if)# endif
Router(config-rpl)# pass
Router(config-rpl)# end-policy
!
Router(config)# route-policy route_policy_2
Router(config-rpl)# if evpn-route-type is 1 then
Router(config-rpl-if)# set extcommunity color color2
Router(config-rpl-if)# endif
Router(config-rpl)# pass
Router(config-rpl)# end-policy
!
```

Configure EVPN route policy for tail-end coloring by exporting EVPN policy to color EVPN type 1 route. At the head-end, ODN uses color advertised by tail-end to create SR-TE tunnel.

```
Router(config)# evpn
Router(config-evpn)# evi 1
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-policy export route_policy_1
```

Configure EVPN route policy for head-end coloring by importing EVPN policy to color EVPN type 1 route at the head-end. ODN uses color configured by head-end to create SR-TE tunnel.

```
Router(config)# evpn
Router(config-evpn)# evi 2
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-policy import route_policy_2
```

Configure EVPN VPWS over SR-TE Policy

This example shows how to configure EVPN VPWS over SR-TE policy.

```
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xg
Router(config-l2vpn-xc)# p2p xc100
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether1.100
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 target 1 source 2
Router(config-l2vpn-xc-p2p-pw)# exit
Router(config-l2vpn-xc-p2p)# exit
Router(config-l2vpn-xc)# p2p xc200
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether1.200
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 2 target 3 source 4
```

Verify EVPN VPWS over SR-TE using ODN

The following show commands display the output for EVPN VPWS PW over SR-TE using ODN configuration.

```
/* View PCE topology on R3 */

Router# show pce ipv4 topology summary
Wed Jul 27 22:00:37.109 UTC

PCE's topology database summary:
--------------------------------

Topology nodes:                3
Prefixes:                      3
Prefix SIDs:
  Total:                       0
  Regular:                     0
  Strict:                      0
Links:
  Total:                      12
  EPE:                         0
Adjacency SIDs:
  Total:                      12
  Unprotected:                12
  Protected:                   0
  EPE:                         0

Private Information:
Lookup Nodes                   0
```

```
Consistent                    no

Update Stats (from IGP and/or BGP):
  Nodes added:                 7
  Nodes deleted:               0
  Links added:                32
  Links deleted:               0
  Prefix added:               47
  Prefix deleted:              0

Topology Ready Summary:
  Ready:                     yes
  PCEP allowed:              yes
  Last HA case:        migration
  Timer value (sec):          40
  Timer:
    Running: no   >>>>> Check if the timer is running.
```

```
/* View PCE configuration on R3. This show command works only on PCE server. */

Router# show pce ipv4 peer
Wed Jul 27 22:02:00.421 UTC

PCE's peer database:
--------------------
Peer address: 1.2.3.4
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation, SRv6

Peer address: 3.4.5.6
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation, SRv6
```

```
/* View PCC configuration on both R1 and R3 */

Router# show segment-routing traffic-eng pcc ipv4 peer
Wed Jul 27 22:01:47.566 UTC

PCC's peer database:
--------------------

Peer address: 3.4.5.6,
  Precedence: 255, (best PCE)
  State up
  Capabilities: Stateful, Update, Segment-Routing, Instantiation, SRv6


Router# show l2vpn xconnect interface Bundle-Ether 1.100 detail
Mon Jul 25 19:19:01.443 UTC

Group xg, XC xc100, state is up; Interworking none
  AC: Bundle-Ether1.100, state is up
  EVPN: neighbor 3.4.5.6, PW ID: evi 1, ac-id 1, state is up ( established )
    XC ID 0xa0000002
    Encapsulation MPLS
    Encap type Ethernet, control word enabled
    Sequencing not set
   Preferred path Active : SR TE srte_c_1_ep_3.4.5.6 (BSID:24021, IFH:0xf000054), On-Demand,
 fallback enabled
    Ignore MTU mismatch: Enabled
    Transmit MTU zero: Enabled
    Tunnel : Up


Router# show segment-routing traffic-eng policy
```

```
Wed Jul 27 22:03:47.253 UTC

SR-TE policy database
---------------------

Color: 1, End-point: 3.4.5.6
  Name: srte_c_1_ep_3.4.5.6
  Status:
    Admin: up  Operational: up for 00:31:53 (since Jul 27 21:31:54.148)
  Candidate-paths:
.........
    Preference: 100 (BGP ODN) (active)
      Requested BSID: dynamic
.........
      Dynamic (pce 3.4.5.6) (valid)
        Metric Type: IGP,   Path Accumulated Metric: 10
          24005 [Adjacency-SID, 42.0.0.2 - 42.0.0.1]
  Attributes:
    Binding SID: 24021
    Forward Class: Not Configured
    Steering labeled-services disabled: no
    Steering BGP disabled: no
    IPv6 caps enable: yes
    Invalidation drop enabled: no
    Max Install Standby Candidate Paths: 0


Router# show segment-routing traffic-eng forwarding policy color 1 endpoint ipv4 3.4.5.6
detail
Wed Jul 27 22:08:09.961 UTC

SR-TE Policy Forwarding database
--------------------------------

Color: 1, End-point: 3.4.5.6
  Name: srte_c_1_ep_3.4.5.6
  Binding SID: 24021
  Active LSP:
    Candidate path:
      Preference: 100 (BGP ODN)
    Local label: 24020
    Segment lists:
      SL[0]:
        Name: dynamic
        Switched Packets/Bytes: 0/0
        Paths:
          Path[0]:
            Outgoing Label: Pop
            Outgoing Interfaces: Bundle-Ether3
            Next Hop: 42.0.0.1
            Switched Packets/Bytes: 0/0
            FRR Pure Backup: No
            ECMP/LFA Backup: No
            Internal Recursive Label: Unlabelled (recursive)
            Label Stack (Top -> Bottom): { Pop }
            Path-id: 1, Weight: 1

  Policy Packets/Bytes Switched: 0/0
```

Call Admission Control for L2VPN P2P Services over Circuit-Style SR-TE Policies

Table 29: Feature History Table

Feature Name	Release Information	Feature Description
Call Admission Control for L2VPN P2P Services over Circuit-Style SR-TE Policies	Release 7.9.1	This feature allows you to configure guaranteed bandwidth for Layer 2 point-to-point (P2P) services steered over Circuit-Style SR-TE policies. This guaranteed bandwidth ensures that a Circuit-Style SR-TE policy has sufficient bandwidth to accommodate a Layer 2 P2P service. At the same time, it prevents a Layer 2 P2P service from being steered over a Circuit-Style SR-TE policy when there is insufficient available bandwidth.

In Layer 2 Virtual Private Network (L2VPN) point-to-point (P2P) services over Circuit-Style SR-TE policies, Call Admission Control (CAC) is used to ensure that the available bandwidth and other network resources are not overloaded by excessive traffic.

While Circuit-Style SR-TE policies are used to steer traffic along specific paths through the network, based on the specific needs of each L2VPN P2P service, CAC is used to ensure that the total bandwidth required by all active L2VPN P2P services on the network does not exceed the available capacity of the network links.

By combining CAC with Circuit-Style SR-TE policies, network administrators can optimize the routing of traffic through the network while ensuring that the network remains within its capacity limits.

Note For information about Circuit-Style SR-TE policies, refer to Circuit-Style SR-TE Policies in the *Segment Routing Configuration Guide for for Cisco 8000 Series Series Routers*.

Call Admission Control (CAC) prevents resource oversubscription in a network. The resources required to enable a service are allocated and reserved before enabling it.

CAC provides the following functionality:

- Ensures that a Circuit-Style SR-TE policy has sufficient bandwidth to accommodate a Layer 2 P2P service.
- Is aware of the total bandwidth associated with a Circuit-Style SR-TE policy, the available bandwidth of the Circuit-Style SR-TE policy considering all L2 P2P services steered over it, and the bandwidth of the L2 P2P service requesting to be admitted into the Circuit-Style SR-TE policy.

- Prevents a L2 P2P service from being steered over a Circuit-Style SR-TE policy when there is insufficient available bandwidth.

Usage Guideline and Limitations

- LDP-signaled L2 P2P services and EVPN VPWS L2 P2P services are supported.
- If a PW is configured with a bandwidth value but is not configured with a preferred path, then the PW stays down with the "admitted bandwidth" set to 0. See L2VPN Preferred path, on page 171.

Configure CAC for L2VPN P2P Services over Circuit-Style SR-TE Policies

To configure CAC for EVPN VPWS L2 P2P services, use the **admission-control bandwidth** *bandwidth* command. The range for *bandwidth* is from 1 to 4294967295 in kbps.

```
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group evpn_vpws
Router(config-l2vpn-xc)# p2p evpn_vpws_1001
Router(config-l2vpn-xc-p2p)# interface TenGigE0/1/0/1.1001
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1001 target 10001 source 20001
Router(config-l2vpn-xc-p2p-pw)# admission-control bandwidth 24000
```

Running Configuration

```
l2vpn
 xconnect group evpn_vpws
  p2p evpn_vpws_1001
   interface TenGigE0/1/0/1.1001
   neighbor evpn evi 1001 target 10001 source 20001
    admission-control bandwidth 24000
   !
  !
 !
!
```

To configure CAC for LDP-signaled L2 P2P services, use the **bandwidth** *bandwidth* command. The range for *bandwidth* is from 1 to 4294967295 in kbps.

```
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xcon1
Router(config-l2vpn-xc)# p2p vplw1002
Router(config-l2vpn-xc-p2p)# interface TenGigE0/0/1/1.1002
Router(config-l2vpn-xc-p2p)# neighbor ipv4 3.3.3.3 pw-id 1002
Router(config-l2vpn-xc-p2p-pw)# bandwidth 24200
```

Running Configuration

```
l2vpn
 xconnect group xcon1
  p2p vplw1002
   interface TenGigE0/0/1/1.1002
   neighbor ipv4 3.3.3.3 pw-id 1002
    bandwidth 24200
   !
  !
 !
!
```

Verification

Use the **show l2vpn cac-db** command to display the total bandwidth of the policy, the available bandwidth, and the reserved bandwidth.

```
Router# show l2vpn cac-db

Policy Name: sr-srte_c_100_ep_3.3.3.3
  Total Bandwidth: 24000
  Available Bandwidth: 11000
  Reserved Bandwidth: 13000
  Service count: 1
  Pseudowire info:
  EVPN/AToM  VPN ID     AC ID      Reqd BW(kbps)   Alloc BW(kbps)  State
  ------------------------------------------------------------------------------
  EVPN     1          1          13000              13000             NOT CONF
```