EVPN MPLS Transport

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

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
EVPN bridging and E-Line services over BGP-LU underlay with SR	Release 25.2.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Modular Systems (8800 [LC ASIC: P100]) You can now configure end-to-end services between data centers using the BGP Labeled Unicast (BGP-LU) underlay with segment routing. The feature supports EVPN E-LAN and E-Line services and enables load balancing across transport, BGP-LU, and service levels using segment routing.
EVPN Bridging and E-Line Services over BGP-LU Underlay	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*) * The EVPN Bridging and E-Line Services over BGP-LU Underlay functionality is now extended to the Cisco 8712-MOD-M routers.

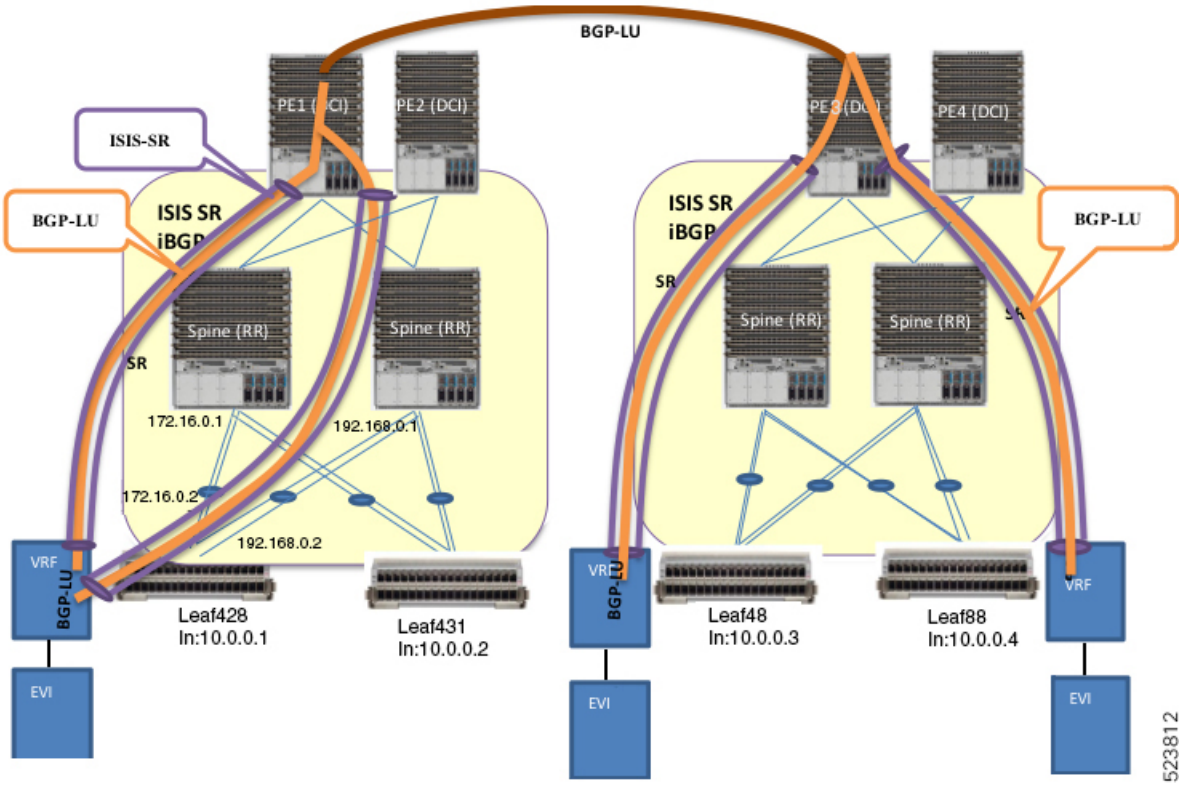
EVPN Bridging and E-Line Services over BGP-LU Underlay	Release 24.3.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) * The EVPN Bridging and E-Line Services over BGP-LU Underlay functionality is now extended to: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
EVPN Bridging and E-Line Services over BGP-LU Underlay	Release 24.2.11	Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*) * The EVPN Bridging and E-Line Services over BGP-LU Underlay functionality is now extended to routers with the 88-LC1-36EH line cards.
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 E-LAN 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 1: 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:

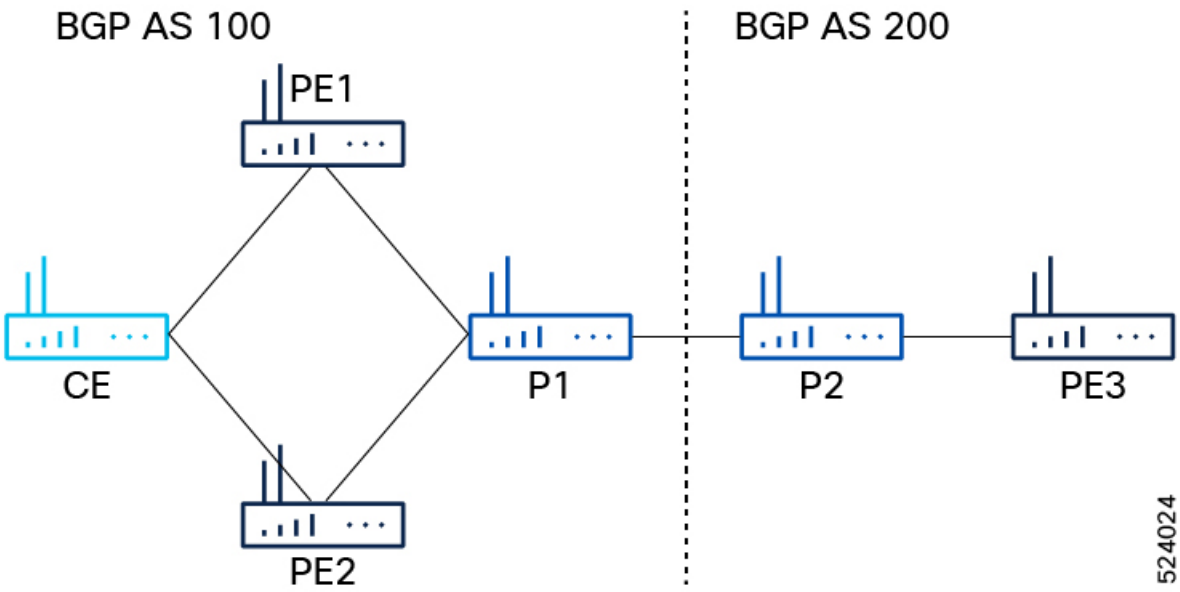
- Consider two data centers that are connected through DCI. Configure EVPN with bridging and inter-subnet routing on the leaf nodes.
- The leaf acts as default gateway for its local hosts.
- Connect hosts to leaf nodes. Leaf nodes are routed across the spines. For DC interconnectivity, the spines are 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.

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

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 2:



- 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
```

Configure EVPN bridging and E-Line services over BGP-LU underlay with SR

Follow the steps to configure EVPN bridging and E-Line services over BGP-LU underlay with SR:

Procedure

Step 1 Configure segment routing with IS-IS to enable SR within the IGP domain and assign prefix SIDs to the router loopback interface.

Configure all the participating nodes, which include the PE or Route Reflector (RR) nodes in the underlay to run IS-IS and advertise their loopbacks with SIDs. The prefix SID varies for each node and instance.

Example:

```
Router#config
Router(config)#router isis igp1
Router(config-isis)#address-family ipv4 unicast
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
```

```
Router(config-isis)#address-family ipv6 unicast
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
Router(config-isis)#interface Loopback 0
Router(config-isis-if)#passive
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-if)#address-family ipv6 unicast
Router(config-isis-if-af)#prefix-sid index 101
```

Step 2 Run the **router bgp** command to configure BGP and router ID on all the nodes.

Ensure that the `bgp router-id` is the loopback address used for peering and SR SID assignment. The router ID varies for each node.

Example:

```
Router(config)#router bgp 65600
Router(config-bgp)#nsr
Router(config-bgp)#bgp router-id 192.168.1.1
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#ibgp policy out enforce-modifications
```

Step 3 Enable the BGP-LU address families and advertise the router loopback address with an associated label derived from the SR node SID.

Configure all the participating nodes to advertise the router loopback address through BGP-LU.

Example:

The network and SID index vary for each node. The following is a sample configuration for IPv4 and IPv6 unicast address families.

```
Router#config
Router(config)#router bgp 64600
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#additional-paths receive
Router(config-bgp-af)#additional-paths send
Router(config-bgp-af)#nexthop trigger-delay critical 50
Router(config-bgp-af)#network 192.168.1.1/32 route-policy SID (1)
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#additional-paths receive
Router(config-bgp-af)#additional-paths send
Router(config-bgp-af)#nexthop trigger-delay critical 50
Router(config-bgp-af)#network 2001:DB8::/32 route-policy SID (101)
Router(config-bgp-af)#allocate-label all
```

Example:

Sample route-policy configuration.

```
Router#config
Router(config)#route-policy SID ($SID)
Router(config-rpl)#set label-index $SID
Router(config-rpl)#end-policy
```

Step 4 Configure the BGP neighbors or neighbor groups to establish iBGP peering between all the participating nodes such as PEs and RRs, and enable the BGP-LU and EVPN address families.

Example:

```
Router(config)#router bgp 64600
Router(config-bgp)#session-group current
Router(config-bgp-sngrp)#remote-as 64600
Router(config-bgp-sngrp)#update-source Loopback 0

/* Configure a BGP neighbor group for BGP-LU for IPv4 and IPv6 address families */
Router(config-bgp)#neighbor-group bgp-lu-rr
Router(config-bgp-nbrgrp)#use session-group current
Router(config-bgp-nbrgrp)#address-family ipv4 labeled-unicast
Router(config-bgp-nbrgrp-af)#next-hop-self
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family ipv6 labeled-unicast
Router(config-bgp-nbrgrp-af)#next-hop-self
Router(config-bgp-nbrgrp-af)#exit

/* Configure BGP neighbor group for L2VPN services such as VPLS-VPWS and EVPN address families */
Router(config-bgp-nbrgrp)#address-family l2vpn vpls-vpws
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family l2vpn evpn

/* Associate an individual BGP neighbor with a predefined neighbor group */
Router(config-bgp)#neighbor 192.168.101.1
Router(config-bgp-nbr)#use neighbor-group bgp-lu-rr
Router(config-bgp-nbr)#exit
```

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 2: Feature History Table

Feature Name	Release Information	Feature Description
VPWS over SR-TE Policy	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*) * The VPWS over SR-TE policy functionality is now extended to the Cisco 8712-MOD-M routers.
VPWS over SR-TE Policy	Release 24.3.1	Introduced in this release on: Fixed Systems (8200, 8700); Modular Systems (8800 [LC ASIC: P100]) (select variants only*) * The VPWS over SR-TE policy functionality is now extended to: • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
VPWS over SR-TE Policy	Release 24.2.11	Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*) * The VPWS over SR-TE policy functionality is now extended to routers with the 88-LC1-36EH line cards.

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.

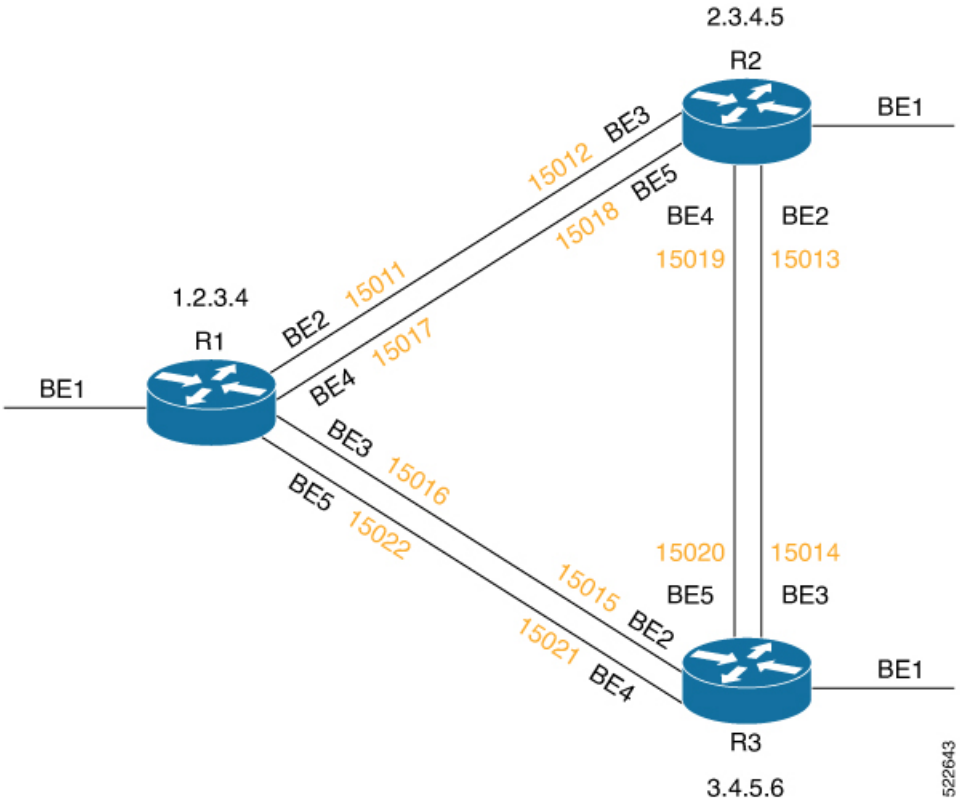
SR-TE uses SR policies to control the path the PW takes through the network, as opposed to using Label Distribution Protocol (LDP) to distribute labels.

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 3: 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.

Decoupled mode for L2VPN and EVPN VPWS services

Decoupled Mode is a configuration for Layer 2 Virtual Private Networks (L2VPN) and Ethernet VPN Virtual Private Wire Services (EVPN VPWS) that:

- enhances network reliability and resilience,
- enables the Pseudowire (PW) to operate independently of the Attachment Circuit (AC), ensuring uninterrupted PW traffic even when the AC fails, and
- uses the xconnect (XC), a virtual connection linking the AC and PW segments, to manage state transitions effectively.

Table 3: Feature History Table

Feature Name	Release Information	Feature Description

Decoupled Mode for L2VPN and EVPN VPWS services	Release 25.2.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100, K100]) Decoupled mode improves fault tolerance by allowing the PE router to maintain the PW in an active state independently of the AC status. Unlike the traditional coupled mode, which requires both AC and PW to be active for traffic flow, decoupled mode ensures uninterrupted PW traffic even during AC failures. The feature introduces these changes: **CLI**: • **decoupled-mode** **YANG Data Model**: • `Cisco-IOS-XR-l2vpn-cfg.yang` • `Cisco-IOS-XR-l2vpn-oper.yang` (see GitHub, YANG Data Models Navigator)

Decoupled mode states

Based on the operational status of the AC and PW, the XC dynamically manages traffic flow by transitioning between these states:

- **Bound state**: When both AC and PW segments are active, the XC is in a bound state, enabling traffic to flow between the AC and PW segments.
- **Unbound state**: When either AC or PW segment experiences a failure, the XC enters an unbound state and drops any pending data on the AC.
  - If the AC segment experiences a failure, the PE router brings the PW connection down and stops the traffic flow or reroutes traffic through alternative paths.
  - If the PW segment experiences a failure, the XC brings the AC connection down and stops the traffic flow.
- **Bound down state**:
  - When AC segment experiences a failure, the XC enters an bound down state where the PE routers keeps the PW active and continue the PW traffic flow between them.
  - If the PW segment experiences a failure, the XC enters an unbound state, brings the AC segment down, and stops the traffic flow.

Before Release 25.2.1, the default configuration for VPWS was the coupled mode, which supported only bound and unbound XC states. For more information, see the VPWS over SR-TE Policy, on page 14.

From Release 25.2.1, you can configure the decoupled mode in addition to the existing coupled mode. Decoupled mode allows PW traffic to continue forwarding even when the AC experiences a failure, enhancing service resilience.

Limitations for decoupled mode

The router does not support decoupled mode for these configurations:

- PWHE Attachment Circuit (AC) segment
- Monitor session AC segment
- Multi-segment Pseudowire (PW)
- Local switching between AC
- BGP Auto-Discovery for PW
- Multi-homing EVPN circuits

How decoupled mode works

Summary

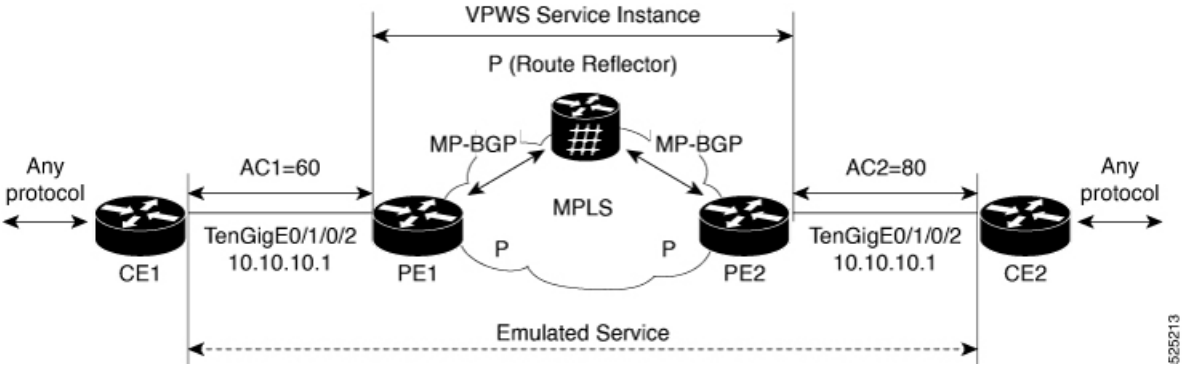
Decoupled Mode enhances network resilience by allowing Provider Edge (PE) routers to manage PW and AC connections independently. This ensures uninterrupted PW traffic and dynamic fault handling between Customer Edge (CE) devices and PE routers.

The key components that are involved in decoupled mode are:

- **Customer edge device**: CE1 and CE2
- **Provider edge router**: PE1 and PE2
- CE1 is connected to PE1
- CE2 is connected to PE2
- PE1 and PE2 are interconnected, and routing information is exchanged through a route reflector.

Workflow

Figure 4: VPWS Traffic flow



Here's how packet transmission takes place in decoupled mode under the different scenarios:

Scenario 1: Both AC and PW are active

When both the AC and PW are active, the XC remains in a bound state, enabling seamless traffic flow between CE1 and CE2. The PE routers perform the following steps:

1. CE1 sends a packet to PE1 through the active AC connection.
2. PE1 receives the packet and performs a lookup to determine that the packet should be forwarded to PE2 over the active PW.
3. PE1 encapsulates the packet and transmits it over the PW to PE2.
4. PE2 receives the packet and performs a lookup to determine the next hop. It forwards the packet to CE2 through the active AC connection.
5. CE2 successfully receives the packet.

Scenario 2: AC experiences a failure

When the AC experiences a failure, the XC transitions to a bound down state, allowing the PW to remain operational and ensuring traffic continues to flow across the PW. The PE routers perform the following steps:

1. CE1 sends a packet to PE1.
   a. PE1 detects the AC failure but does not signal the PW to become inactive, keeping the PW operational.
   b. The XC stays in the bound-down state, allowing traffic to continue flowing across the PW between PE routers despite the AC failure.
2. PE1 receives the packet and performs a lookup to determine that the packet should be forwarded to PE2 over the active PW.
3. PE2 receives the packet and performs a lookup to determine the next hop. It forwards the packet to CE2 through the active AC connection.
4. CE2 successfully receives the packet.

Scenario 3: PW experiences a failure

When the PW experiences a failure, the XC transitions to an unbound state, stopping all traffic flow. The PE routers perform the following steps:

1. CE1 sends a packet to PE1 through the active AC connection.
2. PE1 detects the PW failure and signals the AC to become inactive.
3. The route reflector updates the AC state change to PE2.
4. PE2 blocks traffic from CE2, while PE1 blocks traffic from CE1.
5. The XC drops any pending data on the AC, and traffic flow stops.

Configure decoupled mode for L2VPN and EVPN VPWS services

Perform these steps on PE routers to configure decoupled mode for L2VPN and EVPN VPWS services:

Procedure

Step 1 Enable L2VPN services and configure a point-to-point (p2p) pseudowire service that extends a Layer 2 network across a Layer 3 IP and MPLS network.

Example:

```
Router# configuration
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group g1
Router(config-l2vpn-xc)# p2p xc1
```

Step 2 Enable decoupled mode within p2p services, which keeps the PW active even if the AC experiences a failure.

Example:

```
Router(config-l2vpn-xc-p2p)# decoupled-mode
```

Step 3 Enable interface and configure PW, ipv4 , EVPN or MPLS services for establishing the required L2VPN service over a network.

Example:

```
Router(config-l2vpn-xc-p2p)# interface TenGigE0/1/0/2
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 1 service-id 10011
Router(config-l2vpn-xc-p2p)# commit
```

Step 4 Run **show l2vpn** command to verify the XC, AC, and PW states.

Example:

In this example, both the AC and PW are in the bound state, and the XC is in the up state.

```
Router# show l2vpn xconnect group g1 xc-name p1 detail
Group g1, XC p1, state is up; Interworking none
Decoupled mode: Enabled
  AC: GigabitEthernet0/0/0/2.1, state is up
    Type VLAN; Num Ranges: 1
    Rewrite Tags: []
    VLAN ranges: [1, 1]
    MTU 1504; XC ID 0x1; interworking none
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
      drops: illegal VLAN 0, illegal length 0
  PW: neighbor 1.1.1.1, PW ID 1, state is up ( established )
    PW class not set, XC ID 0xfff80013
    Encapsulation MPLS, protocol LDP
    Source address 2.2.2.2
    PW type Ethernet, control word disabled, interworking none
    PW backup disable delay 0 sec
    Sequencing not set
    Ignore MTU mismatch: Disabled
    Transmit MTU zero: Disabled
    LSP : Up
    Nexthop type: IPV4 1.1.1.1
```

Example:

In this example, due to a local AC fault, the AC is in the down state and the PW is in the up state.

```
Router# show l2vpn xconnect group g1 xc-name p1 detail
```

```
Group g1, XC p1, state is down; Interworking none
Decoupled mode: Enabled
  AC: GigabitEthernet0/0/0/2.1, state is down (Admin)
    Type VLAN; Num Ranges: 1
    Rewrite Tags: []
    VLAN ranges: [1, 1]
    MTU 1504; XC ID 0x1; interworking none
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
      drops: illegal VLAN 0, illegal length 0
  PW: neighbor 1.1.1.1, PW ID 1, state is up ( established )
    PW class not set, XC ID 0xfff80013
    Encapsulation MPLS, protocol LDP
    Source address 2.2.2.2
    PW type Ethernet, control word disabled, interworking none
    PW backup disable delay 0 sec
    Sequencing not set
    Ignore MTU mismatch: Disabled
    Transmit MTU zero: Disabled
    LSP : Up
    Nexthop type: IPV4 1.1.1.1
```

Step 5 Run the **show l2vpn forwarding xconnect <xc-id> detail location <location>** command to verify the xconnect and PW binding state.

Example:

In this example, the XC is in the down state, both AC and PW are in a bound state, and the segment can still carry traffic across the PW despite the AC failure.

```
Router# show l2vpn forwarding xconnect 0x1 detail location 0/0/CPU0

Local interface: GigabitEthernet0/0/0/2.1, Xconnect id: 0x1, Status: down
  Segment 1
    AC, GigabitEthernet0/0/0/2.1, status: Bound
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
  Segment 2
    MPLS, Destination address: 1.1.1.1, pw-id: 1, status: Bound
    Pseudowire label: 24014
    Control word disabled
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
    PD System Data:  0x00000000, 0x00000000, 0x00000000, 0x01000000, 0x22000000,
                     0x00000000, 0x00000000
```

Example:

In this example, the XC is in the up state and both AC and PW are in bound state.

```
Router# show l2vpn forwarding xconnect 0x1 detail location 0/0/CPU0

Local interface: GigabitEthernet0/0/0/2.1, Xconnect id: 0x1, Status: up
  Segment 1
    AC, GigabitEthernet0/0/0/2.1, status: Bound
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
  Segment 2
    MPLS, Destination address: 1.1.1.1, pw-id: 1, status: Bound
```

```
Pseudowire label: 24014
Control word disabled
```

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.

Prerequisites

It is recommended to configure MPLS LDP (Label Distribution Protocol) as the baseline setup to ensure proper label distribution and network stability, particularly when handling scenarios involving unconfiguration and reconfiguration of loopback interfaces.

Configuration Example

```
/* Configure MPLS LDP */
Router(config)# mpls ldp
Router(config-ldp)# router-id 1.2.3.4
```

Attach the policy to the L2VPN instance.

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 4: 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 14.

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
```