Configure Segment Routing over IPv6 (SRv6)

Segment Routing for IPv6 (SRv6) is the implementation of Segment Routing over the IPv6 dataplane.

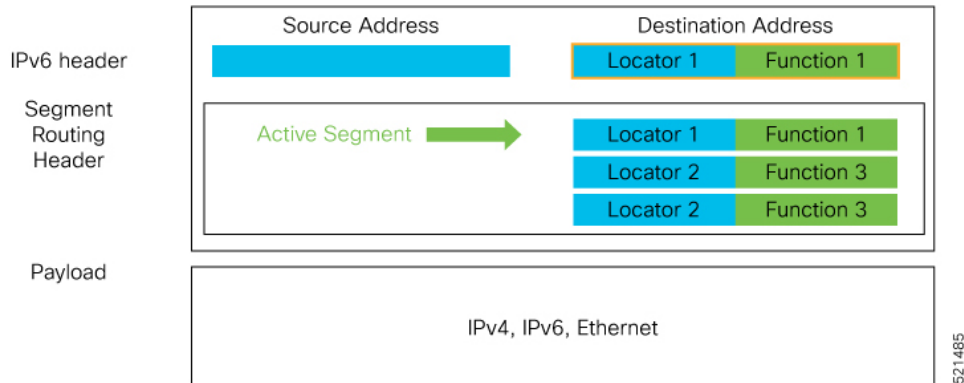## Segment Routing over IPv6 Overview

Segment Routing (SR) can be applied on both MPLS and IPv6 data planes. Segment Routing over IPv6 (SRv6) extends Segment Routing support with IPv6 data plane.

In an SR-MPLS enabled network, an MPLS label represents an instruction. The source nodes programs the path to a destination in the packet header as a stack of labels.

SRv6 introduces the Network Programming framework that enables a network operator or an application to specify a packet processing program by encoding a sequence of instructions in the IPv6 packet header. Each instruction is implemented on one or several nodes in the network and identified by an SRv6 Segment Identifier (SID) in the packet. The SRv6 Network Programming framework is defined in IETF RFC 8986 SRv6 Network Programming.

In SRv6, an IPv6 address represents an instruction. SRv6 uses a new type of IPv6 Routing Extension Header, called the Segment Routing Header (SRH), in order to encode an ordered list of instructions. The active segment is indicated by the destination address of the packet, and the next segment is indicated by a pointer in the SRH.
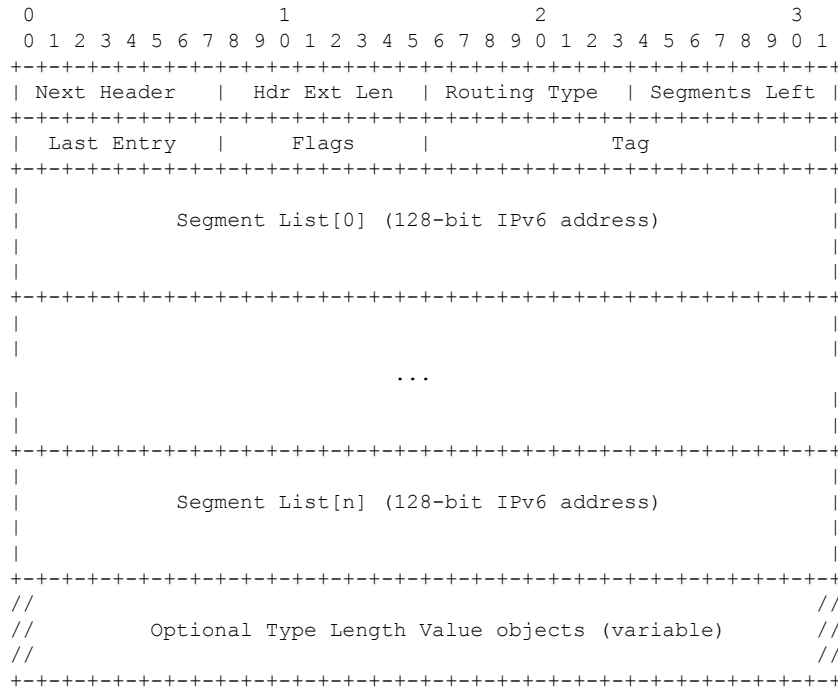
Figure 1: Network Program in the Packet Header

The SRv6 SRH is documented in IETF RFC IPv6 Segment Routing Header (SRH).

The SRH is defined as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128-bit IPv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
                               ...
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128-bit IPv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//          Optional Type Length Value objects (variable)     //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following list explains the fields in SRH:

- Next header—Identifies the type of header immediately following the SRH.

- Hdr Ext Len (header extension length)—The length of the SRH in 8-octet units, not including the first 8 octets.

- Segments left—Specifies the number of route segments remaining. That means, the number of explicitly listed intermediate nodes still to be visited before reaching the final destination.

- Last Entry—Contains the index (zero based) of the last element of the segment list.

- Flags— Contains 8 bits of flags.

- Tag—Tag a packet as part of a class or group of packets like packets sharing the same set of properties.

- Segment list—128-bit IPv6 addresses representing the nth segment in the segment list. The segment list encoding starts from the last segment of the SR policy (path). That means the first element of the segment list (Segment list [0]) contains the last segment of the SR policy, the second element contains the penultimate segment of the SR policy and so on.

In SRv6, a SID represents a 128-bit value, consisting of the following three parts:

- Locator: This is the first part of the SID with most significant bits and represents an address of a specific SRv6 node.

- Function: This is the portion of the SID that is local to the owner node and designates a specific SRv6 function (network instruction) that is executed locally on a particular node, specified by the locator bits.

- Args: This field is optional and represents optional arguments to the function.

The locator part can be further divided into two parts:

- SID Block: This field is the SRv6 network designator and is a fixed or known address space for an SRv6 domain. This is the most significant bit (MSB) portion of a locator subnet.

- Node Id: This field is the node designator in an SRv6 network and is the least significant bit (LSB) portion of a locator subnet.

SRv6 Node Roles

Each node along the SRv6 packet path has a different functionality:

- Source node—A node that can generate an IPv6 packet with an SRH (an SRv6 packet), or an ingress node that can impose an SRH on an IPv6 packet.

- Transit node—A node along the path of the SRv6 packet (IPv6 packet and SRH). The transit node does not inspect the SRH. The destination address of the IPv6 packet does not correspond to the transit node.

- Endpoint node—A node in the SRv6 domain where the SRv6 segment is terminated. The destination address of the IPv6 packet with an SRH corresponds to the end point node. The segment endpoint node executes the function bound to the SID

SRv6 Head-End Behaviors

The SR Headend with Encapsulation behaviors are documented in the IETF RFC 8986 SRv6 Network Programming.

The SR Headend with Insertion head-end behaviors are documented in the following IETF draft:

https://datatracker.ietf.org/doc/draft-filsfils-spring-srv6-net-pgm-insertion/

This section describes a set of SR Policy headend behaviors. The following list summarizes them:

- H.Encaps—SR Headend Behavior with Encapsulation in an SRv6 Policy

- H.Encaps.Red—H.Encaps with Reduced Encapsulation

- H.Insert—SR Headend with insertion of an SRv6 Policy

- H.Insert.Red—H.Insert with reduced insertion

SRv6 Endpoint Behaviors

The SRv6 endpoint behaviors are documented in the IETF RFC 8986 SRv6 Network Programming.

The following is a subset of defined SRv6 endpoint behaviors that can be associated with a SID.

- End—Endpoint function. The SRv6 instantiation of a Prefix SID [RFC8402].

- End.X—Endpoint with Layer-3 cross-connect. The SRv6 instantiation of an Adj SID [RFC8402].

- End.DX6—Endpoint with decapsulation and IPv6 cross-connect (IPv6-L3VPN - equivalent to per-CE VPN label).

- End.DX4—Endpoint with decapsulation and IPv4 cross-connect (IPv4-L3VPN - equivalent to per-CE VPN label).

- End.DT6—Endpoint with decapsulation and IPv6 table lookup (IPv6-L3VPN - equivalent to per-VRF VPN label).

- End.DT4—Endpoint with decapsulation and IPv4 table lookup (IPv4-L3VPN - equivalent to per-VRF VPN label).

- End.DT46—Endpoint with decapsulation and specific IP table lookup (IP-L3VPN - equivalent to per-VRF VPN label).

- End.DX2—Endpoint with decapsulation and L2 cross-connect (L2VPN use-case).

- End.B6.Encaps—Endpoint bound to an SRv6 policy with encapsulation. SRv6 instantiation of a Binding SID.

- End.B6.Encaps.RED—End.B6.Encaps with reduced SRH. SRv6 instantiation of a Binding SID.

SRv6 Endpoint Behavior Variants

Depending on how the SRH is handled, different behavior variants are defined for the End and End.X behaviors. The End and End.X behaviors can support these variants, either individually or in combinations.

- **Penultimate Segment Pop (PSP) of the SRH variant**—An SR Segment Endpoint Nodes receive the IPv6 packet with the Destination Address field of the IPv6 Header equal to its SID address.

  A penultimate SR Segment Endpoint Node is one that, as part of the SID processing, copies the last SID from the SRH into the IPv6 Destination Address and decrements the Segments Left value from one to zero.

  The PSP operation takes place only at a penultimate SR Segment Endpoint Node and does not happen at non-penultimate endpoint nodes. When a SID of PSP-flavor is processed at a non-penultimate SR Segment Endpoint Node, the PSP behavior is not performed since Segments Left would not be zero.

  The SR Segment Endpoint Nodes advertise the SIDs instantiated on them via control plane protocols. A PSP-flavored SID is used by the Source SR Node when it needs to instruct the penultimate SR Segment Endpoint Node listed in the SRH to remove the SRH from the IPv6 header.

- **Ultimate Segment Pop (USP) of the SRH variant**—The SRH processing of the End and End.X behaviors are modified as follows:

  If Segments Left is 0, then:

1. Update the Next Header field in the preceding header to the Next Header value of the SRH

2. Decrease the IPv6 header Payload Length by 8*(Hdr Ext Len+1)

3. Remove the SRH from the IPv6 extension header chain

4. Proceed to process the next header in the packet

One of the applications of the USP flavor is when a packet with an SRH is destined to an application on hosts with smartNICs implementing SRv6. The USP flavor is used to remove the consumed SRH from the extension header chain before sending the packet to the host.

- **Ultimate Segment Decapsulation (USD) variant**—The Upper-layer header processing of the End and End.X behaviors are modified as follows:

  - **End** behavior: If the Upper-layer Header type is 41 (IPv6), then:

    1. Remove the outer IPv6 Header with all its extension headers

    2. Submit the packet to the egress IPv6 FIB lookup and transmission to the new destination

    3. Else, if the Upper-layer Header type is 4 (IPv4)

    4. Remove the outer IPv6 Header with all its extension headers

    5. Submit the packet to the egress IPv4 FIB lookup and transmission to the new destination

    6. Else, process as per Section 4.1.1 (Upper-Layer Header) of IETF RFC 8986 SRv6 Network Programming

  - **End.X** behavior: If the Upper-layer Header type is 41 (IPv6) or 4 (IPv4), then:

    1. Remove the outer IPv6 Header with all its extension headers

    2. Forward the exposed IP packet to the L3 adjacency J

    3. Else, process as per Section 4.1.1 (Upper-Layer Header) of IETF RFC 8986 SRv6 Network Programming

One of the applications of the USD flavor is the case of TI-LFA in P routers with encapsulation with H.Encaps. The USD flavor allows the last Segment Endpoint Node in the repair path list to decapsulate the IPv6 header added at the TI-LFA Point of Local Repair and forward the inner packet.

Usage Guidelines and Limitations

General Guidelines and Limitations

- Cisco IOS XR supports the following SRv6 SID behaviors and variants:

  - END with PSP

  - END.X with PSP

  - END.DT4

  - END.DT6

- SRv6 Underlay support includes:

- IGP redistribution/leaking between levels

- Prefix Summarization on ABR routers

- IS-IS TI-LFA

- Microloop Avoidance

- Flex-algo

Configuring SRv6

To enable SRv6 globally, you should first configure a locator with its prefix. The IS-IS protocol announces the locator prefix in IPv6 network and SRv6 applications (like ISIS, BGP) use it to allocate SIDs.

The following usage guidelines and restrictions apply while configuring SRv6.

- All routers in the SRv6 domain should have the same SID block (network designator) in their locator.

- The locator length should be 64-bits long.

  - The SID block portion (MSBs) cannot exceed 40 bits. If this value is less than 40 bits, user should use a pattern of zeros as a filler.

  - The Node Id portion (LSBs) cannot exceed 24 bits.

- You can configure up to 8 locators to support SRv6 Flexible Algorithm. All locators prefix must share the same SID block (first 40-bits).

Enabling SRv6 with Locator

This example shows how to globally enable SRv6 and configure locator.

```
Router(config)# segment-routing srv6
Router(config-srv6)# locators
Router(config-srv6-locators)# locator myLoc1
Router(config-srv6-locator)# prefix 2001:db8:0:a2::/64
```

Optional: Configuring Encapsulation Parameters

This example shows how to configure encapsulation parameters when configuring SRv6. These optional parameters include:

- Source Address of outer encapsulating IPv6 header: The default source address for encapsulation is one of the loopback addresses.

- The hop limit of outer-encapsulating IPv6 header. The range is from 1 to 254; the default value for hop-limit is 254.

```
Router(config)# segment-routing srv6
Router(config-srv6)# encapsulation source-address 1::1
Router(config-srv6)# hop-limit 60
```

Optional: Enabling Syslog Logging for Locator Status Changes

This example shows how to enable the logging of locator status.

```
Router(config)# segment-routing srv6
Router(config-srv6)# logging locator status
```

Verifying SRv6 Manager

This example shows how to verify the overall SRv6 state from SRv6 Manager point of view. The output displays parameters in use, summary information, and platform specific capabilities.

```
Router# show segment-routing srv6 manager
Parameters:
  Parameters:
  SRv6 Enabled: Yes
  SRv6 Operational Mode:
    Base:
      SID Base Block: 2001:db8::/40
  Encapsulation:
    Source Address:
      Configured: 1::1
      Default: 5::5
    Hop-Limit: Default
    Traffic-class: Default
Summary:
  Number of Locators: 1 (1 operational)
  Number of SIDs: 4 (0 stale)
  Max SIDs: 64000
  OOR:
    Thresholds: Green 3200, Warning 1920
    Status: Resource Available
        History: (0 cleared, 0 warnings, 0 full)
    Block 2001:db8:0:a2::/64:
        Number of SIDs free: 65470
        Max SIDs: 65470
        Thresholds: Green 3274, Warning 1965
        Status: Resource Available
            History: (0 cleared, 0 warnings, 0 full)
Platform Capabilities:
  SRv6: Yes
  TILFA: Yes
  Microloop-Avoidance: Yes
  Endpoint behaviors:
    End (PSP)
    End.X (PSP)
    End.DX6
    End.DX4
    End.DT6
    End.DT4
    End.DX2
    uN (PSP/USD)
    uA (PSP/USD)
    uDT6
    uDT4
    uDX2
    uB6 (Insert.Red)
  Headend behaviors:
    T
    H.Insert.Red
    H.Encaps.Red
  Security rules:
    SEC-1
    SEC-2
    SEC-3
  Counters:
    CNT-1
    CNT-3
  Signaled parameters:
    Max-SL        : 3
```

```
   Max-End-Pop-SRH : 3
   Max-H-Insert    : 3 sids
   Max-H-Encap     : 3 sids
   Max-End-D       : 4
Configurable parameters (under srv6):
  Encapsulation:
    Source Address: Yes
    Hop-Limit     : value=Yes, propagate=No
    Traffic-class : value=Yes, propagate=Yes
Max SIDs: 64000
SID Holdtime: 3 mins
```

Verifying SRv6 Locator

This example shows how to verify the locator configuration and its operational status.

```
Router# show segment-routing srv6 locator myLoc1 detail
Name                 ID      Prefix                   Status
-------------------- ------- ------------------------ -------
myLoc1*              5       2001:db8:0:a2::/64       Up
  (*): is-default
  Interface:
    Name: srv6-myLoc1
    IFH : 0x00000170
    IPv6 address: 2001:db8:0:a2::/64
  Chkpt Obj ID: 0x2fc8
  Created: Apr 25 06:21:57.077 (00:03:37 ago)
```

Verifying SRv6 Local SIDs

This example shows how to verify the allocation of SRv6 local SIDs off locator(s).

```
Router# show segment-routing srv6 locator myLoc1 sid

SID                       Function    Context                        Owner
    State  RW
------------------------- ----------- ------------------------------  ------------------
  -----  --
2001:db8:0:a2:1::         End (PSP)   'default':1                    sidmgr
    InUse  Y
2001:db8:0:a2:40::        End.DT4     'VRF1'                         bgp-100
    InUse  Y
2001:db8:0:a2:41::        End.X (PSP) [Hu0/1/0/1, Link-Local]        isis-srv6
    InUse  Y
```

The following example shows how to display detail information regarding an allocated SRv6 local SID.

```
Router# show segment-routing srv6 locator myLoc1 sid 2001:db8:0:a2:40:: detail

SID                       Function    Context                        Owner
    State  RW
------------------------- ----------- ------------------------------  ------------------
  -----  --
2001:db8:0:a2:40::        End.DT4     'VRF1'                         bgp-100
    InUse  Y
  SID context: { table-id=0xe0000011 ('VRF1':IPv4/Unicast) }
  Locator: myLoc1'
  Allocation type: Dynamic
```

```
Created: Feb  1 14:04:02.901 (3d00h ago)
```

Similarly, you can display SID information across locators by using the **show segment-routing sid** command.

show Commands

You can use the following **show** commands to verify the SRv6 global and locator configuration:

Command	Description
show segment-routing srv6 manager	Displays the summary information from SRv6 manager, including platform capabilities.
show segment-routing srv6 locator *locator-name* [**detail**]	Displays the SRv6 locator information on the router.
show segment-routing srv6 locator *locator-name* **sid** [[*sid-ipv6-address* [**detail**]	Displays the information regarding SRv6 local SID(s) allocated from a given locator.
show segment-routing srv6 sid [*sid-ipv6-address* | **all** | **stale**] [**detail**]	Displays SID information across locators. By default, only "active" (i.e. non-stale) SIDs are displayed.
show route ipv6 local-srv6	Displays all SRv6 local-SID prefixes in IPv6 RIB.

Configuring SRv6 under IS-IS

Intermediate System-to-Intermediate System (IS-IS) protocol already supports segment routing with MPLS dataplane (SR-MPLS). This feature enables extensions in IS-IS to support Segment Routing with IPv6 data plane (SRv6). The extensions include advertising the SRv6 capabilities of nodes and node and adjacency segments as SRv6 SIDs.

SRv6 IS-IS performs the following functionalities:

1. Interacts with SID Manager to learn local locator prefixes and announces the locator prefixes in the IGP domain.

2. Learns remote locator prefixes from other IS-IS neighbor routers and installs the learned remote locator IPv6 prefix in RIB or FIB.

3. Allocate or learn prefix SID and adjacency SIDs, create local SID entries, and advertise them in the IGP domain.

Usage Guidelines and Restrictions

The following usage guidelines and restrictions apply for SRv6 IS-IS:

   • An IS-IS address-family can support either SR-MPLS or SRv6, but both at the same time is not supported.

Configuring SRv6 under IS-IS

To configure SRv6 IS-IS, use the following command:

- **router isis** *instance* **address-family ipv6 unicast segment-routing srv6 locator** *locator* [**level** {**1** | **2**}]—Enable SRv6 under the IS-IS IPv6 address-family and assign SRv6 locator(s) to it. Use the **level** {**1** | **2**} keywords to advertise the locator only in the specified IS-IS level.

The following example shows how to configure SRv6 under IS-IS.

```
Router(config)# router isis core
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# segment-routing srv6
Router(config-isis-srv6)# locator myLoc1 level 1
Router(config-isis-srv6-loc)# exit
```

For more information about configuring IS-IS, refer to the "Implementing IS-IS" chapter in the *Routing Configuration Guide* .

Configuring SRv6 IS-IS TI-LFA

This feature introduces support for implementing Topology-Independent Loop-Free Alternate (TI-LFA) using IS-IS SRv6.

TI-LFA provides link protection in topologies where other fast reroute techniques cannot provide protection. The goal of TI-LFA is to reduce the packet loss that results while routers converge after a topology change due to a link failure. TI-LFA leverages the post-convergence path which is planned to carry the traffic and ensures link and node protection within 50 milliseconds. TI-LFA with IS-IS SR-MPLS is already supported.

Usage Guidelines and Restrictions

The following usage guidelines and restrictions apply:

Configuring SRv6 IS-IS TI-LFA

The following example shows how to configure SRv6 IS-IS TI-LFA.

> **Note** Complete the Configuring SRv6 before performing these steps.

```
Router(config)# router isis core
Router(config-isis)# address-family ipv6 unicast
Router(config-isis-af)# segment-routing srv6
Router(config-isis-srv6)# locator locator1
Router(config-isis-srv6-loc)# exit
Router(config-isis)# interface loopback 0
Router(config-isis-if)# passive
Router(config-isis-if)# address-family ipv6 unicast
Router(config-isis-if-af)# exit
Router(config-isis)# interface bundle-ether 1201
Router(config-isis-if)# address-family ipv6 unicast
Router(config-isis-if-af)# fast-reroute per-prefix
Router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
Router(config-isis-if-af)# exit
Router(config-isis)# interface bundle-ether 1301
Router(config-isis-if)# address-family ipv6 unicast
```

```
Router(config-isis-if-af)# fast-reroute per-prefix
Router(config-isis-if-af)# fast-reroute per-prefix ti-lfa
Router(config-isis-if-af)# exit
```

Verification

This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the **show isis ipv6 fast-reroute** *ipv6-prefix* **detail** command.

```
Router# show isis ipv6 fast-reroute cafe:0:0:66::/64 detail
Thu Nov 22 16:12:51.983 EST

L1 cafe:0:0:66::/64 [11/115] low priority
     via fe80::2, TenGigE0/0/0/6, SRv6-HUB6, Weight: 0
      Backup path: TI-LFA (link), via fe80::1, Bundle-Ether1201 SRv6-LF1, Weight: 0, Metric:
 51
          P node: SRv6-TP8.00 [8::8], SRv6 SID: cafe:0:0:88:1:: End (PSP)
          Backup-src: SRv6-HUB6.00
        P: No, TM: 51, LC: No, NP: No, D: No, SRLG: Yes
     src SRv6-HUB6.00-00, 6::6
```

This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the **show route ipv6** *ipv6-prefix* **detail** command.

```
Router# show route ipv6 cafe:0:0:66::/64 detail
Thu Nov 22 16:14:07.385 EST

Routing entry for cafe:0:0:66::/64
  Known via "isis srv6", distance 115, metric 11, type level-1
  Installed Nov 22 09:24:05.160 for 06:50:02
  Routing Descriptor Blocks
    fe80::2, from 6::6, via TenGigE0/0/0/6, Protected
      Route metric is 11
      Label: None
      Tunnel ID: None
      Binding Label: None
      Extended communities count: 0
      Path id:1        Path ref count:0
      NHID:0x2000a(Ref:11)
      NHID eid:0xffffffffffffffff
      SRv6 Headend: H.Insert.Red [base], SRv6 SID-list {cafe:0:0:88:1::}
      Backup path id:65
    fe80::1, from 6::6, via Bundle-Ether1201, Backup (TI-LFA)
      Repair Node(s): 8::8
      Route metric is 51
      Label: None
      Tunnel ID: None
      Binding Label: None
      Extended communities count: 0
      Path id:65              Path ref count:1
      NHID:0x2000d(Ref:11)
      NHID eid:0xffffffffffffffff
      SRv6 Headend: H.Insert.Red [base], SRv6 SID-list {cafe:0:0:88:1::}
      MPLS eid:0x1380800000001
```

This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the **show cef ipv6** *ipv6-prefix* **detail location** *location* command.

```
Router# show cef  ipv6 cafe:0:0:66::/64 detail location 0/0/cpu0
Thu Nov 22 17:01:58.536 EST
cafe:0:0:66::/64, version 1356, SRv6 Transit, internal 0x1000001 0x2 (ptr 0x8a4a45cc) [1],
 0x0 (0x8a46ae20), 0x0 (0x8c8f31b0)
 Updated Nov 22 09:24:05.166
 local adjacency fe80::2
 Prefix Len 64, traffic index 0, precedence n/a, priority 2
  gateway array (0x8a2dfaf0) reference count 4, flags 0x500000, source rib (7), 0 backups
                [5 type 3 flags 0x8401 (0x8a395d58) ext 0x0 (0x0)]
  LW-LDI[type=3, refc=1, ptr=0x8a46ae20, sh-ldi=0x8a395d58]
  gateway array update type-time 1 Nov 22 09:24:05.163
 LDI Update time Nov 22 09:24:05.163
 LW-LDI-TS Nov 22 09:24:05.166
   via fe80::2/128, TenGigE0/0/0/6, 8 dependencies, weight 0, class 0, protected [flags
0x400]
    path-idx 0 bkup-idx 1 NHID 0x2000a [0x8a2c2fd0 0x0]
    next hop fe80::2/128
   via fe80::1/128, Bundle-Ether1201, 8 dependencies, weight 0, class 0, backup (TI-LFA)
[flags 0xb00]
    path-idx 1 NHID 0x2000d [0x8c2670b0 0x0]
    next hop fe80::1/128, Repair Node(s): 8::8
    local adjacency
     SRv6 H.Insert.Red SID-list {cafe:0:0:88:1::}

    Load distribution: 0 (refcount 5)

    Hash  OK  Interface                 Address
    0     Y   TenGigE0/0/0/6            fe80::2
```

This example shows how to verify the SRv6 IS-IS TI-LFA configuration using the **show cef ipv6
fast-reroute-db** command.

```
Router# show cef ipv6 fast-reroute-db
Sun Dec  9 20:23:08.111 EST

 PROTECT-FRR: per-prefix [1, 0x0, 0x0, 0x98c83270]
 protect-interface: Te0/0/0/6 (0x208)
 protect-next-hop:  fe80::2/128
 ipv6 nhinfo [0x977397d0]
 Update Time Dec  9 17:29:42.427

    BACKUP-FRR: per-prefix [5, 0x0, 0x2, 0x98c83350]
    backup-interface: BE1201 (0x800002c)
    backup-next-hop:  fe80::1/128
    ipv6 nhinfo [0x977396a0 protect-frr: 0x98c83270]
 Update Time Dec  9 17:29:42.428

 PROTECT-FRR: per-prefix [1, 0x0, 0x0, 0x98c830b0]
 protect-interface: BE1201 (0x800002c)
 protect-next-hop:  fe80::1/128
 ipv6 nhinfo [0x977396a0]
 Update Time Dec  9 17:29:42.429

    BACKUP-FRR: per-prefix [5, 0x0, 0x1, 0x98c83190]
    backup-interface: Te0/0/0/6 (0x208)
    backup-next-hop:  fe80::2/128
    ipv6 nhinfo [0x977397d0 protect-frr: 0x98c830b0]
 Update Time Dec  9 17:29:42.429
```

SRv6 Services: IPv4 L3VPN

The SRv6-based IPv4 L3VPN feature enables deployment of IPv4 L3VPN over a SRv6 data plane. Traditionally, it was done over an MPLS-based system. SRv6-based L3VPN uses SRv6 Segment IDs (SIDs) for service segments instead of labels. SRv6-based L3VPN functionality interconnects multiple sites to resemble a private network service over public infrastructure. To use this feature, you must configure SRv6-base.

For this feature, BGP allocates an SRv6 SID from the locator space, configured under SRv6-base and VPNv4 address family. For more information on this, refer . The BGP SID can be allocated in the following ways:

- Per-VRF mode that provides End.DT4 support. End.DT4 represents the Endpoint with decapsulation and IPv4 table lookup.

- Per-CE mode that provides End.DX4 cross connect support. End.DX4 represents the Endpoint with decapsulation and IPv4 cross-connect.

BGP encodes the SRv6 SID in the prefix-SID attribute of the IPv4 L3VPN Network Layer Reachability Information (NLRI) and advertises it to IPv6 peering over an SRv6 network. The Ingress PE (provider edge) router encapsulates the VRF IPv4 traffic with the SRv6 VPN SID and sends it over the SRv6 network.

Usage Guidelines and Limitations

- SRv6 locator can be assigned globally, for all VRFs, or for an individual VRF.

- Equal-Cost Multi-path (ECMP) and Unequal Cost Multipath (UCMP) are supported.

- BGP, OSPF, Static are supported as PE-CE protocol.

- MPLS L3VPN and SRv6 L3VPN interworking gateway is not supported.

Configuring SRv6 based IPv4 L3VPN

To enable SRv6-based L3VPN, you need to configure SRv6 under BGP and configure the SID allocation mode. The following example shows how to configure SRv6-based L3VPN:

Configure an Individual VRF with Per-VRF Label Allocation Mode

```
RP/0/0/CPU0:Router(config-bgp-af)# vrf vrf1
RP/0/0/CPU0:Router(config-bgp-vrf)# rd 106:1
RP/0/0/CPU0:Router(config-bgp-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:Router(config-bgp-vrf-af)# segment-routing srv6
RP/0/0/CPU0:Router(config-bgp-vrf-af-srv6)# alloc mode per-vrf
RP/0/0/CPU0:Router(config-bgp-vrf-af-srv6)# exit
RP/0/0/CPU0:Router(config-bgp-vrf-af)# exit
RP/0/0/CPU0:Router(config-bgp-vrf)# neighbor 10.1.2.2
RP/0/0/CPU0:Router(config-bgp-vrf-nbr)# remote-as 100
RP/0/0/CPU0:Router(config-bgp-vrf-nbr)# address-family ipv4 unicast
```

Configure an Individual VRF with Per-CE Label Allocation Mode

```
RP/0/0/CPU0:Router(config-bgp-af)# vrf vrf2
RP/0/0/CPU0:Router(config-bgp-vrf)# rd 106:2
RP/0/0/CPU0:Router(config-bgp-vrf)# address-family ipv4 unicast
RP/0/0/CPU0:Router(config-bgp-vrf-af)# segment-routing srv6
RP/0/0/CPU0:Router(config-bgp-vrf-af-srv6)# alloc mode per-ce
```

```
RP/0/0/CPU0:Router(config-bgp-vrf-af-srv6)# exit
RP/0/0/CPU0:Router(config-bgp-vrf-af)# exit
RP/0/0/CPU0:Router(config-bgp-vrf)# neighbor 10.1.2.2
RP/0/0/CPU0:Router(config-bgp-vrf-nbr)# remote-as 100
RP/0/0/CPU0:Router(config-bgp-vrf-nbr)# address-family ipv4 unicast
```

Verification

The following example shows how to verify the SRv6 based L3VPN configuration using the **show segment-routing srv6 sid** command.

In this example, End.X represents Endpoint function with Layer-3 cross-connect, End.DT4 represents Endpoint with decapsulation and IPv4 table lookup, and End.DX4 represents Endpoint with decapsulation and IPv4 cross-connect.

```
 RP/0/0/CPU0:SRv6-Hub6# show segment-routing srv6 sid
*** Locator: 'my_locator' ***
SID                         Function    Context                        Owner
    State  RW
-------------------------  ----------  -----------------------------   -----------------
  -----  --
cafe:0:0:66:1::            End (PSP)    'my_locator':1                 sidmgr
    InUse  Y
cafe:0:0:66:40::           End.X (PSP)  [Te0/0/0/2, Link-Local]        isis-srv6
    InUse  Y
cafe:0:0:66:41::           End.X (PSP)  [BE6801, Link-Local]           isis-srv6
    InUse  Y
cafe:0:0:66:42::           End.X (PSP)  [BE5601, Link-Local]           isis-srv6
    InUse  Y
cafe:0:0:66:43::           End.X (PSP)  [BE5602, Link-Local]           isis-srv6
    InUse  Y
cafe:0:0:66:44::           End.DT4      'VRF1'                         bgp-100
    InUse  Y
cafe:0:0:66:45::           End.DT4      'VRF2'                         bgp-100
    InUse  Y
cafe:0:0:66:46::           End.DX4      'VRF2':3                       bgp-100
    InUse  Y
cafe:0:0:66:47::           End.DX4      'VRF2':4                       bgp-100
    InUse  Y
```

The following example shows how to verify the SRv6 based L3VPN configuration using the **show segment-routing srv6** *SID-prefix* **detail** command.

```
RP/0/RP0/CPU0:SRv6-Hub6# show segment-routing srv6 sid cafe:0:0:66:44:: detail
Sun Dec  9 16:52:54.015 EST
*** Locator: 'my_locator' ***
SID                         Function    Context                        Owner
    State  RW
-------------------------  ----------  -----------------------------   -----------------
  -----  --
cafe:0:0:66:44::           End.DT4      'VRF1'                         bgp-100
    InUse  Y
  SID context: { table-id=0xe0000001 ('VRF1':IPv4/Unicast) }
  Locator: 'my_locator'
  Allocation type: Dynamic
  Created: Dec  8 16:34:32.506 (1d00h ago)

RP/0/RP0/CPU0:SRv6-Hub6# show segment-routing srv6 sid cafe:0:0:66:47:: detail
Sun Dec  9 16:54:26.073 EST
*** Locator: 'my_locator' ***
SID                         Function    Context                        Owner
```

```
     State  RW
------------------------ ---------- ----------------------------   ------------------
     ----- --
cafe:0:0:66:47::                End.DX4         'VRF2':4                      bgp-100
    InUse  Y
  SID context: { table-id=0xe0000002 ('VRF2':IPv4/Unicast), nh-set-id=4 }
  Locator: 'my_locator'
  Allocation type: Dynamic
  Created: Dec  9 16:49:44.714 (00:04:41 ago)
```

The following example shows how to verify the SRv6 based L3VPN configuration using the **show bgp vpnv4 unicast rd***route-distinguisher/prefix* command on Egress PE.

```
RP/0/RP0/CPU0:SRv6-Hub6# show bgp vpnv4 unicast rd 106:1 10.15.0.0/30
Wed Nov 21 16:08:44.765 EST
BGP routing table entry for 10.15.0.0/30, Route Distinguisher: 106:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker            2282449     2282449
    SRv6-VPN SID: cafe:0:0:66:44::/128
Last Modified: Nov 21 15:50:34.235 for 00:18:10
Paths: (2 available, best #1)
  Advertised to peers (in unique update groups):
    2::2
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    2::2
  200
    10.1.2.2 from 10.1.2.2  (10.7.0.1)
      Origin IGP, localpref 200, valid, internal, best, group-best, import-candidate
      Received Path ID 0, Local Path ID 1, version 2276228
      Extended community: RT:201:1
  Path #2: Received by speaker 0
  Not advertised to any peer
  200
    10.2.2.2 from 10.2.2.2 (10.20.1.2)
      Origin IGP, localpref 100, valid, internal
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: RT:201:1
```

The following example shows how to verify the SRv6 based L3VPN configuration using the **show bgp vpnv4 unicast rd***route-distinguisher prefix* command on Ingress PE.

```
RP/0/RP0/CPU0:SRv6-LF1# show bgp vpnv4 unicast rd 106:1 10.15.0.0/30
Wed Nov 21 16:11:45.538 EST
BGP routing table entry for 10.15.0.0/30, Route Distinguisher: 106:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker            2286222     2286222
Last Modified: Nov 21 15:47:26.288 for 00:24:19
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  200, (received & used)
    6::6 (metric 24) from 2::2 (6.6.6.6)
      Received Label 3
      Origin IGP, localpref 200, valid, internal, best, group-best, import-candidate,
not-in-vrf
      Received Path ID 1, Local Path ID 1, version 2286222
      Extended community: RT:201:1
      Originator: 6.6.6.6, Cluster list: 2.2.2.2
      SRv6-VPN-SID: T1-cafe:0:0:66:44:: [total 1]
```

The following example shows how to verify the SRv6 based L3VPN configuration using the **show route vrf***vrf-name/prefix***detail** command.

```
RP/0/RP0/CPU0:SRv6-LF1# show route vrf VRF1 10.15.0.0/30 detail
Wed Nov 21 16:35:17.775 EST
Routing entry for 10.15.0.0/30
  Known via "bgp 100", distance 200, metric 0
  Tag 200, type internal
  Installed Nov 21 16:35:14.107 for 00:00:03
  Routing Descriptor Blocks
    6::6, from 2::2
      Nexthop in Vrf: "default", Table: "default", IPv6 Unicast, Table Id: 0xe0800000
      Route metric is 0
      Label: None
      Tunnel ID: None
      Binding Label: None
      Extended communities count: 0
      Source RD attributes: 0x0000:106:1
      NHID:0x0(Ref:0)
      SRv6 Headend: H.Encaps.Red [base], SID-list { cafe:0:0:66:44:: }
      MPLS eid:0x1380600000001
  Route version is 0xd (13)
  No local label
  IP Precedence: Not Set
  QoS Group ID: Not Set
  Flow-tag: Not Set
  Fwd-class: Not Set
  Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_REMOTE
  Download Priority 3, Download Version 3038384
  No advertising protos.
```

The following example shows how to verify the SRv6 based L3VPN configuration for per-ce allocation mode using the **show bgp vrf***vrf-name***nexthop-set** command.

```
RP/0/RP0/CPU0:SRv6-Hub6# show bgp vrf VRF2 nexthop-set
Wed Nov 21 15:52:17.464 EST
 Resilient per-CE nexthop set, ID 3
 Number of nexthops 1, Label 0, Flags 0x2200
 SRv6-VPN SID: cafe:0:0:66:46::/128
 Nexthops:
10.1.2.2
 Reference count 1,
 Resilient per-CE nexthop set, ID 4
 Number of nexthops 2, Label 0, Flags 0x2100
 SRv6-VPN SID: cafe:0:0:66:47::/128
 Nexthops:
 10.1.2.2
 10.2.2.2
Reference count 2,
```

The following example shows how to verify the SRv6 based L3VPN configuration using the **show cef vrf***vrf-name prefix* **detail location***line-card* command.

```
RP/0/RP0/CPU0:SRv6-LF1# show cef vrf VRF1 10.15.0.0/30 detail location 0/0/cpu0
Wed Nov 21 16:37:06.894 EST
151.1.0.0/30, version 3038384, SRv6 Transit, internal 0x5000001 0x0 (ptr 0x9ae6474c) [1],
0x0 (0x0), 0x0 (0x8c11b238)
 Updated Nov 21 16:35:14.109
 Prefix Len 30, traffic index 0, precedence n/a, priority 3
  gateway array (0x8cd85190) reference count 1014, flags 0x2010, source rib (7), 0 backups
                [1 type 3 flags 0x40441 (0x8a529798) ext 0x0 (0x0)]
```

```
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Nov 21 14:47:26.816
 LDI Update time Nov 21 14:52:53.073
  Level 1 - Load distribution: 0
  [0] via cafe:0:0:66::/128, recursive
   via cafe:0:0:66::/128, 7 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x8acb53cc 0x0]
    next hop VRF - 'default', table - 0xe0800000
    next hop cafe:0:0:66::/128 via cafe:0:0:66::/64
     SRv6 H.Encaps.Red SID-list {cafe:0:0:66:44::}
    Load distribution: 0 (refcount 1)
    Hash  OK  Interface               Address
    0     Y   Bundle-Ether1201        fe80::2
```

SRv6 Services: IPv4 L3VPN Active-Standby Redundancy using Port-Active Mode

The Segment Routing IPv6 (SRv6) Services: IPv4 L3VPN Active-Standby Redundancy using Port-Active Mode feature provides all-active per-port load balancing for multihoming. The forwarding of traffic is determined based on a specific interface rather than per-flow across multiple Provider Edge routers. This feature enables efficient load-balancing and provides faster convergence. In an active-standby scenario, the active PE router is detected using designated forwarder (DF) election by modulo calculation and the interface of the standby PE router brought down. For Modulo calculation, byte 10 of the Ethernet Segment Identifier (ESI) is used.

Usage Guidelines and Restrictions

- This feature can only be configured for bundle interfaces.

- When an EVPN Ethernet Segment (ES) is configured with port-active load-balancing mode, you cannot configure ACs of that bundle on bridge-domains with a configured EVPN instance (EVI). EVPN Layer 2 bridging service is not compatible with port-active load-balancing.

SRv6 Services for L3VPN Active-Standby Redundancy using Port-Active Mode: Operation

Under port-active operational mode, EVPN Ethernet Segment (ES) routes are exchanged across BGP for the routers servicing the multihomed ES. Each PE router then builds an ordered list of the IP addresses of all PEs connected to the ES, including itself, and assigns itself an ordinal for its position in the list. The ordinals are used with the modulo calculation to determine which PE will be the Designated Forwarder (DF) for a given ES. All non-DF PEs will take the respective bundles out of service.

In the case of link or port failure, the active DF PE withdraws its ES route. This re-triggers DF election for all PEs that service the ES and a new PE is elected as DF.

Configure SRv6 Services L3VPN Active-Standby Redundancy using Port-Active Mode

This section describes how you can configure SRv6 services L3VPN active-standby redundancy using port-active mode under an Ethernet Segment (ES).

Configuration Example

```
/* Configure Ethernet Link Bundles */
Router# configure
Router(config)# interface Bundle-Ether10
Router(config-if)# ipv4 address 10.0.0.2 255.255.255.0
Router(config-if)# ipv6 address 2001:DB8::1
Router(config-if)# lacp period short
Router(config-if)# mac-address 1.2.3
Router(config-if)# bundle wait-while 0
Router(config-if)# exit
Router(config)# interface GigabitEthernet 0/2/0/5
Router(config-if)# bundle id 14 mode active
Router(config-if)# commit

/* Configure load balancing. */
Router# configure
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether10
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 11.11.11.11.11.11.11.11.14
Router(config-evpn-ac-es)# load-balancing-mode port-active
Router(config-evpn-ac-es)# commit
!
/* Configure address family session in BGP. */
Router# configure
Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 192.168.0.2
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 192.168.0.3
Router(config-bgp-nbr)# remote-as 200
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
Router(config-bgp-nbr)# commit
```

Running Configuration

```
interface Bundle-Ether14
 ipv4 address 14.0.0.2 255.255.255.0
 ipv6 address 14::2/64
 lacp period short
 mac-address 1.2.3
 bundle wait-while 0
!
interface GigabitEthernet0/2/0/5
 bundle id 14 mode active
!
evpn
 interface Bundle-Ether14
  ethernet-segment
   identifier type 0 11.11.11.11.11.11.11.11.14
   load-balancing-mode port-active
```

```
  !
 !
!
router bgp 100
 bgp router-id 192.168.0.2
 address-family l2vpn evpn
 !
 neighbor 192.168.0.3
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
 !
!
```

Verification

Verify the SRv6 services L3VPN active-standby redundancy using port-active mode configuration.

```
/* Verify ethernet-segment details on active DF router */
Router# show evpn ethernet-segment interface Bundle-Ether14 detail
Ethernet Segment Id      Interface                         Nexthops
---------------------- --------------------------------- --------------------
0011.1111.1111.1111.1114 BE14                             192.168.0.2
                                                          192.168.0.3
    ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether14
     Interface MAC  : 0001.0002.0003
     IfHandle       : 0x000041d0
     State          : Up
     Redundancy     : Not Defined
  ESI type          : 0
     Value          : 11.1111.1111.1111.1114
  ES Import RT       : 1111.1111.1111 (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH
     Configured     : Port-Active
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Peering Details   :
     192.168.0.2 [MOD:P:00]
     192.168.0.3 [MOD:P:00]

  Service Carving Results:
     Forwarders     : 0
     Permanent      : 0
     Elected        : 0
     Not Elected    : 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   : None
  Remote SHG labels : 0


/* Verify bundle Ethernet configuration on active DF router */
Router# show bundle bundle-ether 14
Bundle-Ether14
  Status:                                  Up
```

```
Local links <active/standby/configured>:    1 / 0 / 1
Local bandwidth <effective/available>:       1000000 (1000000) kbps
MAC address (source):                        0001.0002.0003 (Configured)
Inter-chassis link:                          No
Minimum active links / bandwidth:            1 / 1 kbps
Maximum active links:                        64
Wait while timer:                            Off
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
------------------- --------------- ----------- --------------- ----------
Gi0/2/0/5           Local           Active      0x8000, 0x0003    1000000
    Link is Active



/*  Verify ethernet-segment details on standby DF router */
Router# show evpn ethernet-segment interface bundle-ether 10 detail

Router# show evpn ethernet-segment interface Bundle-Ether24 detail
Ethernet Segment Id      Interface                          Nexthops
----------------------- ---------------------------------- --------------------
0011.1111.1111.1111.1114 BE24                               192.168.0.2
                                                            192.168.0.3

  ES to BGP Gates  : Ready
  ES to L2FIB Gates : Ready
  Main port        :
    Interface name : Bundle-Ether24
    Interface MAC  : 0001.0002.0003
    IfHandle       : 0x000041b0
    State          : Standby
    Redundancy     : Not Defined
  ESI type         : 0
    Value          : 11.1111.1111.1111.1114
  ES Import RT      : 1111.1111.1111 (from ESI)
  Source MAC       : 0000.0000.0000 (N/A)
  Topology         :
    Operational    : MH
    Configured     : Port-Active
  Service Carving  : Auto-selection
    Multicast      : Disabled
  Peering Details   :
    192.168.0.2 [MOD:P:00]
    192.168.0.3 [MOD:P:00]

  Service Carving Results:
    Forwarders     : 0
    Permanent      : 0
    Elected        : 0
    Not Elected    : 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   : None
  Remote SHG labels : 0

/* Verify bundle configuration on standby DF router */
Router# show bundle bundle-ether 24

Bundle-Ether24
  Status:                                 LACP OOS (out of service)
  Local links <active/standby/configured>:  0 / 1 / 1
  Local bandwidth <effective/available>:    0 (0) kbps
  MAC address (source):                     0001.0002.0003 (Configured)
  Inter-chassis link:                       No
  Minimum active links / bandwidth:         1 / 1 kbps
  Maximum active links:                     64
  Wait while timer:                         Off
  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
  -------------------- --------------- ----------- -------------- ----------
  Gi0/0/0/4            Local           Standby     0x8000, 0x0002   1000000
      Link is in standby due to bundle out of service state
```