



SRv6 Layer 3 VPN Services and Global Routing

Segment Routing over IPv6 (SRv6) provides a scalable, flexible, and programmable framework for traffic engineering and service delivery within modern network architectures. This chapter functions as a foundational guide, detailing the principles and implementation procedures for core Layer 3 VPN (L3VPN) and global routing services leveraging SRv6.

- [SRv6 Layer 3 Services, on page 1](#)
- [SRv6 BGP-Based Services, on page 2](#)
- [IPv4 L3VPNs over SRv6, on page 5](#)
- [IPv6 L3VPN services over SRv6 networks, on page 21](#)
- [IPv4 BGP Global SRv6 services, on page 33](#)
- [Configure BGP global IPv4 Over SRv6 with Per-CE SID allocation mode, on page 34](#)
- [Configure per-VRF-46 label allocation mode, on page 34](#)
- [IPv6 BGP global SRv6 services, on page 36](#)
- [SRv6 double recursion for multilayer BGP underlay, on page 42](#)

SRv6 Layer 3 Services

SRv6 Layer 3 services are a category of advanced IP routing and VPN capabilities that

- utilize SRv6 SIDs to steer traffic within a network
- enable scalable, flexible, and integrated IP routing solutions by encapsulating payloads with an outer IPv6 header addressed to an SRv6 Service SID, and
- facilitate seamless interconnection of Provider Edge (PE) nodes and VPN formation.

These services support IPv4 and IPv6 L3VPNs, BGP signaling, redundancy modes, VRF-to-VRF route leaking, and dual-stack deployment in mixed IPv4 and IPv6 environments.

Key features of SRv6 L3 services include:

- Support for dual-stack L3VPN services (IPv4 and IPv6) enabling flexible deployment in mixed IP environments.
- Use of BGP for signaling SRv6 Service SIDs, supporting iBGP, eBGP, OSPF, and static PE-CE protocols.
- Support for Equal-Cost Multi-Path (ECMP) and Unequal Cost Multipath (UCMP) routing.
- Interworking capabilities between MPLS L3VPN and SRv6 L3VPN services.

- Mechanisms for VRF-to-VRF route leaking within the SRv6 core.
- Path Maximum Transmission Unit (MTU) discovery for SRv6 encapsulated packets to optimize packet forwarding and avoid fragmentation.

SRv6 BGP-Based Services

The BGP in SRv6 Services is a routing protocol extension that

- enables the signaling and distribution of SRv6 SIDs
- supports advanced VPN and routing services over an SRv6 network, and
- facilitates the encapsulation and interconnection of provider edge routers using SRv6 service SIDs.

Table 1: Feature History Table

Feature Name	Release	Description
VRF Allocation Mode for uDT46 Endpoint Behavior	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Support for uDT46 SRv6 Endpoint Behavior	Release 24.4.1	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*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release	Description
Support for uDT46 SRv6 Endpoint Behavior	Release 7.11.1 Release 7.5.3	This feature adds support for the “Endpoint with decapsulation and specific IP table lookup” SRv6 end-point behavior (uDT46). The End.DT46 behavior is used for dual-stack L3VPNs. This behavior is equivalent to the single per-VRF VPN label (for IPv4 and IPv6) in MPLS.
VRF Allocation Mode for uDT46 Endpoint Behavior	Release 7.11.1	This feature introduces a new VRF allocation mode for uDT46 SIDs for the following BGP-based services: • IPv4 Layer-3 VPNs • IPv6 Layer-3 VPNs • IPv4 BGP global • IPv6 BGP global • L3 EVPN This allocation mode allows for both IPv4/IPv6 address families, and VPNv4/v6 address families, to use a single SID. When this allocation mode is configured under an address family, CE-learned routes, redistributed routes, aggregated routes, local routes, and imported routes will use uDT46 SID when advertised to remote peers. The feature introduces these changes: CLI: • The per-vrf-46 allocation mode is introduced in the following commands: • router bgp as-number address-family {ipv4 ipv6} unicast segment-routing srv6 alloc mode per-vrf-46 • router bgp as-number address-family {vpn4 vpn6} unicast vrf all segment-routing srv6 alloc mode per-vrf-46 • router bgp as-number vrf WORD address-family {ipv4 ipv6} unicast segment-routing srv6 alloc mode per-vrf-46

Extending BGP for SRv6-based overlay services

Building on the messages and procedures defined in IETF draft [BGP/MPLS IP Virtual Private Networks \(VPNs\)](#) BGP has been extended to provide services over an SRv6 network, such as:

- IPv4 Layer-3 VPNs
- IPv6 Layer-3 VPNs

- IPv4 BGP global
- IPv6 BGP global
- Layer-2 VPNs - Ethernet VPNs (EVPN)

For more information about BGP, refer to the Implementing BGP chapter in the *Routing Configuration Guide for Cisco 8000 Series Routers*.

In SRv6-based services, the egress PE signals an SRv6 service SID with the BGP service route. The ingress PE encapsulates the payload in an outer IPv6 header where the destination address is the SRv6 Service SID advertised by the egress PE. BGP messages between PEs carry SRv6 service SIDs as a means to interconnect PEs and form VPNs. SRv6 service SID refers to a segment identifier associated with one of the SRv6 service-specific behaviors advertised by the egress PE router, such as:

- uDT4 (Endpoint with decapsulation and IPv4 table lookup)
- uDT6 (Endpoint with decapsulation and IPv6 table lookup)
- uDT46 (Endpoint with decapsulation and specific IP table lookup)
- uDX4 (Endpoint with decapsulation and IPv4 cross-connect)
- uDX6 (Endpoint with decapsulation and IPv6 cross-connect)

Based on the messages and procedures defined in IETF draft [SRv6 BGP based Overlay services](#), BGP encodes the SRv6 Service SID in the prefix-SID attribute of the corresponding BGP Network Layer Reachability Information (NLRI) and advertises it to its IPv6 BGP peers.

Usage guidelines for SRv6 BGP-based services

- These SRv6 BGP-based services are supported:
 - IPv4 Layer-3 VPNs
 - IPv6 Layer-3 VPNs
 - IPv4 BGP global
 - IPv6 BGP global
- uDT4 and uDT6 for L3VPN and BGP global are supported.
- uDT4, uDT6, and uDT46 for L3VPN and BGP global are supported.
- Dual-Stack L3 Services (IPv4 L3VPN, IPv6 L3VPN, IPv4 BGP global, IPv6 BGP global) are supported.

SRv6 Locator Inheritance Rules

SRv6 locators can be assigned at different levels inside the BGP routing process. BGP allocates SRv6 service SIDs from configured locator spaces according to these inheritance rules:

1. Use the locator as defined under the service.
If not defined under the specific service, then:
2. Use the locator as defined under the corresponding address-family.

If not defined under the corresponding address-family, then:

3. Use the locator as defined globally under BGP.

Configure SRv6 BGP-based services

Procedure

- Step 1** Use the **router bgp as-number segment-routing srv6** command to enable SRv6 globally under the BGP routing process. The *as-number* is from 1-65535.

Example:

```
Node1(config)# router bgp 100 segment-routing srv6
```

- Step 2** Use the **router bgp as-number segment-routing srv6 locator WORD** command to assign an SRv6 locator globally under the BGP routing process.

The *as-number* is from 1-65535.

Example:

```
Node1(config)# router bgp 100 segment-routing srv6 locator Node1-locator
```

IPv4 L3VPNs over SRv6

IPv4 L3VPNs over SRv6 is an SRv6 L3VPN service that

- provides Layer 3 VPN connectivity specifically for IPv4 traffic
- utilizes an SRv6 network as the underlying transport, and
- enables enterprise IPv4 internet connectivity.

This service allows operators to establish Layer 3 VPNs for IPv4 traffic using a SRv6 infrastructure. It leverages advanced routing capabilities to ensure efficient and scalable connectivity, particularly for enterprise environments requiring internet access through VPNs. It supports VPNv4 services, including route leaking between Global Routing Tables (GRT) and Virtual Routing and Forwarding (VRF) instances.

SRv6 VPN BGP route leaking

SRv6 VPN BGP route leaking is a BGP routing mechanism that

- enables the controlled exchange of routing information between Global Routing Tables (GRT) and Virtual Routing and Forwarding (VRF) instances within an SRv6 network
- facilitates VPNv4 services, and
- is essential for providing enterprise IPv4 internet connectivity by allowing specific routes to be shared across different routing contexts.

Table 2: Feature History Table

Feature Name	Release	Description
SRv6 VPN BGP Route Leaking	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
SRv6 VPN BGP Route Leaking	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
SRv6 VPN BGP Route Leaking	Release 7.8.1	This feature supports SRv6 VPN Route-leaking between Global Routing Table (GRT) and Virtual Routing and Forwarding (VRF). This enables Enterprise IPv4 internet connectivity.

This service allows operators to establish Layer 3 VPNs for IPv4 traffic using a SRv6 infrastructure. It leverages advanced routing capabilities to ensure efficient and scalable connectivity, particularly for enterprise environments requiring internet access through VPNs. It supports VPNv4 services, including route leaking between Global Routing Tables (GRT) and Virtual Routing and Forwarding (VRF) instances.

Key concepts of SRv6 VPN BGP route leaking

- **Global Routing Table (GRT):** The Global Routing Table (GRT) is the main routing table within a router, containing all routes that are not associated with a specific VPN or VRF. It is used for forwarding traffic that is not part of any VPN.
- **Virtual Routing and Forwarding (VRF) Instance:** A VRF instance is a virtual router within a physical router, allowing multiple independent routing tables to coexist on the same device. Each VRF instance maintains its own routing table, forwarding table, and interfaces, providing logical separation for different VPNs or customer networks.

Per-VRF-46 allocation mode

Per-VRF-46 allocation mode is an SRv6 SID allocation mode that

- is configured within BGP under an address family
- assigns a specific uDT46 SID to various types of routes when advertised to remote peers, and
- aggregates multiple routes into a single entity for forwarding decisions, improving network efficiency and scalability.

Table 3: Feature History Table

Feature Name	Release	Description
Per-Prefix SRv6 Locator Assignment	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Per-Prefix SRv6 Locator Assignment	Release 24.4.1	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*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Per-Prefix SRv6 Locator Assignment	Release 7.8.1	This feature allows you to assign a specific SRv6 locator for a given prefix or a set of prefixes (IPv4/IPv6 GRT, IPv4/IPv6 VPN). The egress PE advertises the prefix with the specified locator. This allows for per-prefix steering into desired transport behaviors, such as Flex Algo.

Dual-stack L3VPN services for SRv6 Micro-SID on IPv4 and IPv6

Dual-stack L3VPN services is an SRv6 L3VPN capability that

- supports dual-stack (VPNv4/VPNv6) VRFs
- enables SRv6 L3VPN service for both IPv4 (uDT4) and IPv6 (uDT6) on the same interface, sub-interface, or VRF, and
- allows operators to simultaneously and independently access IPv4 and IPv6 without protocol translation, ensuring high processing efficiency and zero information loss.

Table 4: Feature History Table

Feature Name	Release	Description
Dual-Stack L3VPN Services (IPv4, IPv6) (SRv6 Micro-SID)	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Dual-Stack L3VPN Services (IPv4, IPv6) (SRv6 Micro-SID)	Release 24.4.1	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*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release	Description
Dual-Stack L3VPN Services (IPv4, IPv6) (SRv6 Micro-SID)	Release 7.8.1	This feature introduces support for Dual-stack (VPNv4/VPNv6) VRFs. VPNv4/VPNv6 Dual-stack supports both IPv4 (uDT4) and IPv6 (uDT6) based SRv6 L3VPN service on the same interface, sub-interface, or VRF. Dual stacking allows operators to access both IPv4 and IPv6 simultaneously and independent of each other. It avoids the need to translate between two protocol stacks. This results in high processing efficiency and zero information loss.

This feature is designed to provide comprehensive L3VPN connectivity over an SRv6 network for environments that require seamless handling of both IPv4 and IPv6 traffic.

Configure SRv6-based IPv4 L3VPN

To enable SRv6-based L3VPN services for IPv4 traffic by configuring SRv6 under BGP, specifying locators, and defining SID allocation modes.

SRv6-based L3VPNs allow for flexible and scalable IPv4 VPN services over an SRv6 network. The assignment of SRv6 locators and SID allocation modes can be configured at various levels within the BGP configuration.

Procedure

Step 1 Assign an SRv6 locator under BGP.

- Use Case 1: Assigning SRv6 locator globally.

This example shows how to enable SRv6 and configure the SRv6 locator name under BGP Global:

```

Node1(config)# router bgp 100
Node1(config-bgp)# segment-routing srv6
Node1(config-bgp-gbl-srv6)# locator Node1-locator
Node1(config-bgp-gbl-srv6)# exit
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# commit

```

Running configuration

```

router bgp 100
  segment-routing srv6
    locator Node1-locator
  !
  address-family vpnv4 unicast
  !
  neighbor 3001::1:1:1:4
    remote-as 100
    address-family vpnv4 unicast
  !
  !
  vrf vrf_cust1
    rd 100:1
    address-family ipv4 unicast
  !
  !
end

```

- Use Case 2: Assigning SRv6 locator for all VRFs.

This example shows how to enable SRv6 and configure the SRv6 locator for all VRFs under VPNv4 Address Family, with per-VRF label allocation mode:

```

Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# commit

```

Running configuration

```

router bgp 100
  address-family vpnv4 unicast
    vrf all
      segment-routing srv6
        locator Node1-locator
        alloc mode per-vrf
    !
  !
  !
  neighbor 3001::1:1:1:4
    remote-as 100
    address-family vpnv4 unicast
  !
  !
  vrf vrf_cust1
    rd 100:1

```

```

    address-family ipv4 unicast
    !
    !
    !
end

```

- Use Case 3: Assigning SRv6 Locator for a specific VRF

To configure the SRv6 locator for a specific VRF under IPv4 Address Family and specify the allocation mode, use the following commands:

This example shows how to configure the SRv6 locator for an individual VRF, with per-VRF label allocation mode:

```

Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf
Node1(config-bgp-vrf-af-srv6)# commit

```

Running Config

```

router bgp 100
 address-family vpnv4 unicast
    !
 neighbor 3001::1:1:1:4
    remote-as 100
 address-family vpnv4 unicast
    !
    !
 vrf vrf_cust1
    rd 100:1
    address-family ipv4 unicast
        segment-routing srv6
            locator Node1-locator
            alloc mode per-vrf
    !
    !
    !
end

```

This example shows how to configure the SRv6 locator for an individual VRF, with per-CE label allocation mode:

```

Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast

```

```

Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-ce
Node1(config-bgp-vrf-af-srv6)# commit

```

Running Config

```

router bgp 100
 address-family vpnv4 unicast
 !
 neighbor 3001::1:1:1:4
  remote-as 100
  address-family vpnv4 unicast
 !
 !
vrf vrf_cust1
 rd 100:1
 address-family ipv4 unicast
  segment-routing srv6
  locator Node1-locator
  alloc mode per-ce
 !
 !
 !
 !

```

- Use Case 4: Assigning SRv6 Locator for a specific prefix

This use case provides the ability to assign a specific SRv6 locator for a given prefix or a set of prefixes. The egress PE advertises the prefix with the specified locator. This allows for per-prefix steering into desired transport behaviors, such as Flex Algo.

The following example shows a route policy specifying the SID allocation mode with destination-based match:

```

Node1(config)# route-policy set_per_prefix_locator_rpl
Node1(config-rpl)# if destination in (1.1.1.0/24) then
Node1(config-rpl-if)# set srv6-alloc-mode per-vrf locator locator1
Node1(config-rpl-if)# elseif destination in (2.2.2.0/24) then
Node1(config-rpl-elseif)# set srv6-alloc-mode per-vrf locator locator2
Node1(config-rpl-elseif)# elseif destination in (3.3.3.0/24) then
Node1(config-rpl-elseif)# set srv6-alloc-mode per-vrf
Node1(config-rpl-elseif)# elseif destination in (4.4.4.0/24) then
Node1(config-rpl-elseif)# set srv6-alloc-mode per-ce
Node1(config-rpl-elseif)# else
Node1(config-rpl-else)# drop
Node1(config-rpl-else)# endif
Node1(config-rpl)# end-policy
Node1(config)#

```

To assign an SRv6 locator for a specific prefix, configure a route policy to specify the SID allocation mode based on match criteria. Examples of match criteria are destination-based match or community-based match.

- Supported SID allocation modes are per-VRF and per-CE.
- For per-VRF allocation mode, you can also specify the SRv6 locator.
 - If an SRv6 locator is specified in the route policy, BGP will use that to allocate per-VRF SID. If the specified locator is invalid, the SID will not be allocated.
 - If an SRv6 locator is not specified in the route policy, the default locator configured under BGP is used to allocate the SID. If the default locator is not configured, then the SID will not be allocated.

- Per-CE allocation mode always uses the default locator configured under BGP to allocate the SID.

Step 2 Verify that the local and received SIDs have been correctly allocated under VPNv4 and specific VRF (vrf_cust1).

Example:

Model# **show bgp vpnv4 unicast local-sids**

```
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 50
BGP NSR Initial initsync version 18 (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	Local Sid	Alloc mode	Locator
Route Distinguisher: 8:8			
*>i8.8.8.8/32	NO SRv6 Sid	-	-
* i	NO SRv6 Sid	-	-
Route Distinguisher: 1.1.1.1:0 (default for vrf vrf_cust1)			
> 1.1.1.0/24	fc00:0:1:40::	per-vrf	locator1
> 2.2.2.0/24	fc00:8:1:40::	per-vrf	locator2
> 3.3.3.0/24	fc00:9:1:40::	per-vrf	locator4
> 4.4.4.0/24	fc00:9:1:41::	per-ce	locator4
> 1.1.1.5/32	NO SRv6 Sid	-	-
> 3.3.3.3/32	NO SRv6 Sid	-	-
>i8.8.8.8/32	NO SRv6 Sid	-	-

Model# **show bgp vpnv4 unicast received-sids**

```
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 50
BGP NSR Initial initsync version 18 (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	Received Sid
Route Distinguisher: 8:8		
*>i8.8.8.8/32	10.1.1.2	fc00:0:2:42::
* i	2400:2020:42:2fff::1	fc00:0:2:42::
Route Distinguisher: 1.1.1.1:0 (default for vrf vrf_cust1)		
> 1.1.1.0/24	11.1.1.2	NO SRv6 Sid
> 2.2.2.0/24	11.1.1.2	NO SRv6 Sid
> 3.3.3.0/24	11.1.1.2	NO SRv6 Sid
> 4.4.4.0/24	11.1.1.2	NO SRv6 Sid
> 1.1.1.5/32	11.1.1.2	NO SRv6 Sid
> 3.3.3.3/32	13.2.2.2	NO SRv6 Sid
>i8.8.8.8/32	10.1.1.2	fc00:0:2:42::

Configure per-VRF-46 label allocation mode

```

Node1# show bgp vrf vrf_cust1 local-sids
BGP VRF vrf_cust1, state: Active
BGP Route Distinguisher: 1.1.1.1:0
VRF ID: 0x60000004
BGP router identifier 1.1.1.1, local AS number 1
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000013 RD version: 37
BGP main routing table version 37
BGP NSR Initial initsync version 18 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Local Sid                      Alloc mode   Locator
Route Distinguisher: 1.1.1.1:0 (default for vrf vrf_cust1)
*> 1.1.1.0/24        fc00:0:1:40::                per-vrf      locator1
*> 2.2.2.0/24        fc00:8:1:40::                per-vrf      locator2
*> 3.3.3.0/24        fc00:9:1:40::                per-vrf      locator4
*> 4.4.4.0/24        fc00:9:1:41::                per-ce      locator4
*> 1.1.1.5/32        NO SRv6 Sid                  -            -
*> 3.3.3.3/32        NO SRv6 Sid                  -            -
*>i8.8.8.8/32        NO SRv6 Sid                  -            -

Node1# show bgp vrf vrf_cust1 received-sids
BGP VRF vrf_cust1, state: Active
BGP Route Distinguisher: 1.1.1.1:0
VRF ID: 0x60000004
BGP router identifier 1.1.1.1, local AS number 1
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000013 RD version: 37
BGP main routing table version 37
BGP NSR Initial initsync version 18 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop                      Received Sid
Route Distinguisher: 1.1.1.1:0 (default for vrf vrf_cust1)
*> 1.1.1.0/24        11.1.1.2                      NO SRv6 Sid
*> 2.2.2.0/24        11.1.1.2                      NO SRv6 Sid
*> 3.3.3.0/24        11.1.1.2                      NO SRv6 Sid
*> 4.4.4.0/24        11.1.1.2                      NO SRv6 Sid
*> 1.1.1.5/32        11.1.1.2                      NO SRv6 Sid
*> 3.3.3.3/32        13.2.2.2                      NO SRv6 Sid
*>i8.8.8.8/32        10.1.1.2                      fc00:0:2:42::

```

Configure per-VRF-46 label allocation mode

Procedure

Assign SRv6 locator

- Enable SRv6 and configure the SRv6 locator for all VRFs under VPNv4/v6 address family, with per-VRF-46 label allocation mode:

```

Node1(config)# router bgp 200
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# commit

```

- Configure the SRv6 locator for an individual VRF with per-VRF label allocation mode:

```

Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf
Node1(config-bgp-vrf-af-srv6)# commit

```

- Configure the SRv6 locator for an individual VRF for both IPv4 and IPv6 address families, with per-VRF-46 label allocation mode:

```

Node1(config)# router bgp 200
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1

```

Verify the SRv6 based L3VPN configuration

```

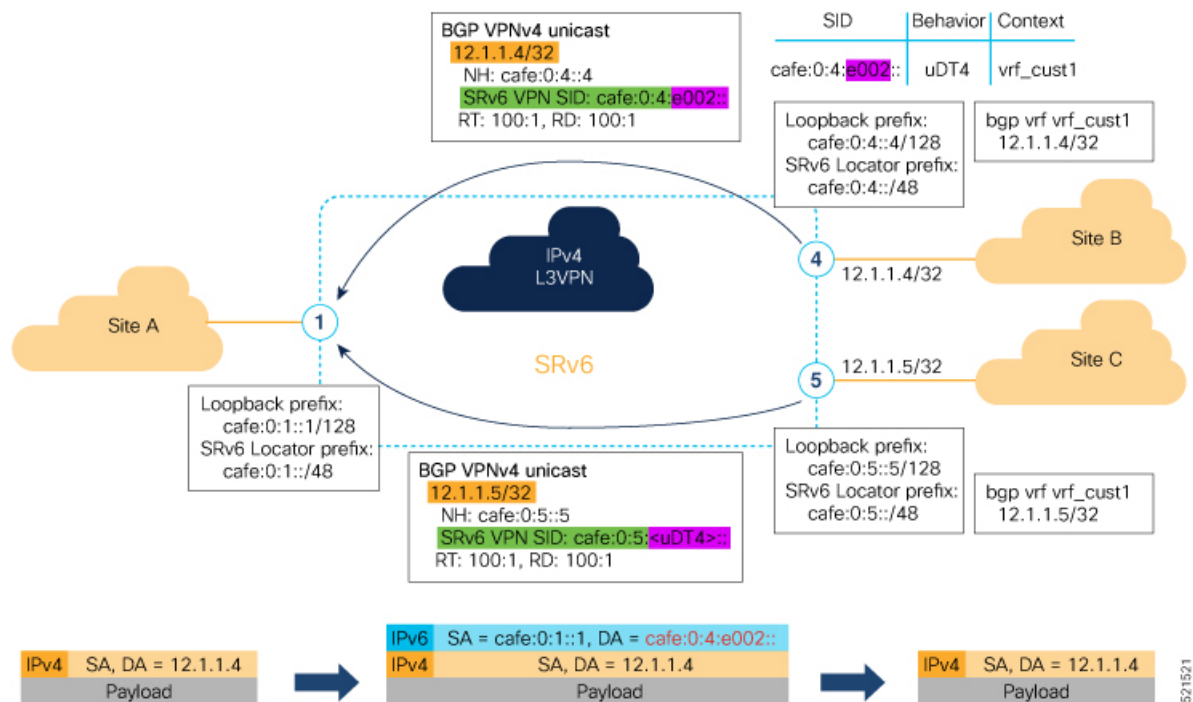
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-vrf-af-srv6)# exit
Node1(config-bgp-vrf-af)# exit
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-vrf-af-srv6)# commit

```

Verify the SRv6 based L3VPN configuration

To confirm that the SRv6-based L3VPN services for IPv4 traffic are correctly configured and operating as expected. This task helps ensure that SRv6 locators, SID allocation modes, and routing information are properly established.

This verification task is typically performed after configuring SRv6-based IPv4 L3VPNs. The examples provided in this task illustrate a common VPNv4 scenario where router Node1 acts as the Ingress Provider Edge (PE), and routers Node4 and Node5 function as Egress PEs.



Procedure

Step 1 Verify the SRv6 SID information using the **show segment-routing srv6 sid** command.

Example:

In this example, we can observe the uDT4 SIDs associated with the IPv4 L3VPN; where uDT4 behavior represents Endpoint with decapsulation and IPv4 table lookup, and uDX4 represents Endpoint with decapsulation and IPv4 cross-connect.

```
Node1# show segment-routing srv6 sid
```

```
*** Locator: 'Node1-locator' ***
```

SID State RW	Behavior	Context	Owner
-----	-----	-----	-----
----- --			
cafe:0:1:: InUse Y	uN (PSP/USD)	'default':1	sidmgr
cafe:0:1:e000:: InUse Y	uA (PSP/USD)	[Hu0/0/0/0, Link-Local]:0	isis-1
cafe:0:1:e001:: InUse Y	uA (PSP/USD)	[Hu0/0/0/1, Link-Local]:0	isis-1
cafe:0:1:e002:: InUse Y	uDT4	'vrf_cust1'	bgp-100
cafe:0:1:e003:: InUse Y	uDT4	'vrf_cust2'	bgp-100
cafe:0:1:e004:: InUse Y	uDT4	'vrf_cust3'	bgp-100
cafe:0:1:e005:: InUse Y	uDT4	'vrf_cust4'	bgp-100
cafe:0:1:e006:: InUse Y	uDT4	'vrf_cust5'	bgp-100

Step 2 Verify the detailed SRv6 SID information for a specific SID using the **show segment-routing srv6SID-prefixdetail** command.

Example:

```
Node1# show segment-routing srv6 sid cafe:0:1:e002:: detail
```

```
Tue Feb 9 17:50:40.621 UTC
```

```
*** Locator: 'Node1-locator' ***
```

SID State RW	Behavior	Context	Owner
-----	-----	-----	-----
----- --			
cafe:0:1:e002:: InUse Y	uDT4	'vrf_cust1'	bgp-100
SID Function: 0xe002 SID context: { table-id=0xe0000011 ('vrf_cust1':IPv4/Unicast) } Locator: 'Node1-locator' Allocation type: Dynamic Created: Feb 9 17:41:07.475 (00:09:33 ago)			

Step 3 Verify the BGP VPNv4 unicast summary using the **show bgp vpnv4 unicast** commands on Egress PE.

Example:

```
Node1# show bgp vpnv4 unicast summary
```

```
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 36
```

Verify the SRv6 based L3VPN configuration

```
BGP NSR Initial initsync version 16 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
```

BGP is operating in STANDALONE mode.

Process	RcvTblVer	bRIB/RIB	LabelVer	ImportVer	SendTblVer	StandbyVer
Speaker	36	36	36	36	36	0

Neighbor	Spk	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	St/PfxRcd
cafe:0:4::4	0	100	47	48	36	0	0	00:40:05	5
cafe:0:5::5	0	100	47	47	36	0	0	00:39:56	5

Node1# **show bgp vpnv4 unicast rd 100:1**

```
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 36
BGP NSR Initial initsync version 16 (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: 100:1 (default for vrf vrf_cust1)
*> 12.1.1.1/32      0.0.0.0                0         32768 ?
*>i12.4.4.4/32      cafe:0:4::4            0        100    0 ?
*>i12.5.5.5/32      cafe:0:5::5            0        100    0 ?
```

Processed 3 prefixes, 3 paths

Node1# **show bgp vpnv4 unicast rd 100:1 12.4.4.4/32**

```
BGP routing table entry for 12.4.4.4/32, Route Distinguisher: 100:1
Versions:
   Process          bRIB/RIB    SendTblVer
   Speaker          22         22
Last Modified: Feb 23 22:57:56.756 for 00:40:08
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local, (received & used)
    cafe:0:4::4 (metric 30) from cafe:0:4::4 (1.1.1.4)
    Received Label 0xe00400
    Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best, import-candidate,
    imported
    Received Path ID 0, Local Path ID 1, version 22
    Extended community: RT:1:1 RT:100:1
    PSID-Type:L3, SubTLV Count:1
    SubTLV:
      T:1(Sid information), Sid:cafe:0:4::, Behavior:63, SS-TLV Count:1
      SubSubTLV:
        T:1(Sid structure):
        Source AFI: VPNv4 Unicast, Source VRF: vrf_cust1, Source Route Distinguisher: 100:1
```

Step 4 Verify the BGP VPNv4 unicast routes for a specific Route Distinguisher (RD) using the **show bgp vpnv4 unicast rdroute-distinguisher prefix** command on Ingress PE.

Example:

Step 5 Verify the SRv6 based L3VPN configuration using the **show route vrf** commands.

Example:

```
Node1# show route vrf vrf_cust1
```

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

```
Gateway of last resort is not set
```

```
L 12.1.1.1/32 is directly connected, 00:44:43, Loopback100
B 12.4.4.4/32 [200/0] via cafe:0:4::4 (nexthop in vrf default), 00:42:45
B 12.5.5.5/32 [200/0] via cafe:0:5::5 (nexthop in vrf default), 00:42:45
```

```
Node1# show route vrf vrf_cust1 12.4.4.4/32
```

```
Routing entry for 12.4.4.4/32
Known via "bgp 100", distance 200, metric 0, type internal
Installed Feb 23 22:57:56.746 for 00:43:12
Routing Descriptor Blocks
  cafe:0:4::4, from cafe:0:4::4
    Nexthop in Vrf: "default", Table: "default", IPv6 Unicast, Table Id: 0xe0800000
    Route metric is 0
No advertising protos.
```

```
Node1# show route vrf vrf_cust1 12.4.4.4/32 detail
```

```
Routing entry for 12.4.4.4/32
Known via "bgp 100", distance 200, metric 0, type internal
Installed Feb 23 22:57:56.746 for 00:43:37
Routing Descriptor Blocks
  cafe:0:4::4, from cafe:0:4::4
    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: 0x0000:100:1
    NHID:0x0(Ref:0)
    SRv6 Headend: H.Encaps.Red [f3216], SID-list {cafe:0:4:e004::}
Route version is 0x1 (1)
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 3
```

Verify the SRv6 based L3VPN configuration

No advertising protos.

Step 6 Verify CEF information for a VRF using the **show cef vrf** commands.

Example:

```
Node1# show cef vrf vrf_cust1
```

Prefix	Next Hop	Interface
0.0.0.0/0	drop	default handler
0.0.0.0/32	broadcast	
12.1.1.1/32	receive	Loopback100
12.4.4.4/32	cafe:0:4::/128	<recursive>
12.5.5.5/32	cafe:0:5::/128	<recursive>
224.0.0.0/4	0.0.0.0/32	
224.0.0.0/24	receive	
255.255.255.255/32	broadcast	

```
Node1# show cef vrf vrf_cust1 12.4.4.4/32
```

```
12.4.4.4/32, version 3, SRv6 Headend, internal 0x5000001 0x30 (ptr 0x78b9a61c) [1], 0x0 (0x0), 0x0 (0x88873720)
Updated Feb 23 22:57:56.749
Prefix Len 32, traffic index 0, precedence n/a, priority 3
via cafe:0:4::/128, 3 dependencies, recursive [flags 0x6000]
path-idx 0 NHID 0x0 [0x78e2da14 0x0]
next hop VRF - 'default', table - 0xe0800000
next hop cafe:0:4::/128 via cafe:0:4::/48
SRv6 H.Encaps.Red SID-list {cafe:0:4:e004::}
```

```
Node1# show cef vrf vrf_cust1 12.4.4.4/32 detail
```

```
12.4.4.4/32, version 3, SRv6 Headend, internal 0x5000001 0x30 (ptr 0x78b9a61c) [1], 0x0 (0x0), 0x0 (0x88873720)
Updated Feb 23 22:57:56.749
Prefix Len 32, traffic index 0, precedence n/a, priority 3
gateway array (0x88a740a8) reference count 5, flags 0x2010, source rib (7), 0 backups
[1 type 3 flags 0x48441 (0x789cbcc8) ext 0x0 (0x0)]
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
gateway array update type-time 1 Feb 23 22:57:56.749
LDI Update time Feb 23 22:57:56.754

Level 1 - Load distribution: 0
[0] via cafe:0:4::/128, recursive

via cafe:0:4::/128, 3 dependencies, recursive [flags 0x6000]
path-idx 0 NHID 0x0 [0x78e2da14 0x0]
next hop VRF - 'default', table - 0xe0800000
next hop cafe:0:4::/128 via cafe:0:4::/48
SRv6 H.Encaps.Red SID-list {cafe:0:4:e004::}

Load distribution: 0 1 (refcount 1)

Hash OK Interface Address
0 Y HundredGigE0/0/0/1 remote
1 Y HundredGigE0/0/0/0 remote
```

Step 7 Verify the BGP prefix information for VRF instances using the **show bgp vrf** commands:

Example:

```

Node1# show bgp vrf vrf_cust1 ipv4 unicast

BGP VRF vrf_cust1, state: Active
BGP Route Distinguisher: 100:1
VRF ID: 0x60000002
BGP router identifier 1.1.1.1, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000011 RD version: 32
BGP main routing table version 36
BGP NSR Initial initsync version 16 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network             Next Hop             Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf vrf_cust1)
*> 12.1.1.1/32          0.0.0.0                      0         32768 ?
*>i12.4.4.4/32          cafe:0:4::4                   0         100   0 ?
*>i12.5.5.5/32          cafe:0:5::5                   0         100   0 ?

Processed 3 prefixes, 3 paths

Node1# show bgp vrf vrf_cust1 ipv4 unicast 12.4.4.4/32
Tue Feb 23 23:39:57.499 UTC
BGP routing table entry for 12.4.4.4/32, Route Distinguisher: 100:1
Versions:
  Process                bRIB/RIB   SendTblVer
  Speaker                 22        22
Last Modified: Feb 23 22:57:56.756 for 00:42:01
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local, (received & used)
    cafe:0:4::4 (metric 30) from cafe:0:4::4 (1.1.1.4)
    Received Label 0xe00400
    Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best, import-candidate,
imported
    Received Path ID 0, Local Path ID 1, version 22
    Extended community: RT:1:1 RT:100:1
    PSID-Type:L3, SubTLV Count:1
    SubTLV:
      T:1(Sid information), Sid:cafe:0:4::, Behavior:63, SS-TLV Count:1
      SubSubTLV:
        T:1(Sid structure):
          Source AFI: VPNv4 Unicast, Source VRF: vrf_cust1, Source Route Distinguisher: 100:1

```

IPv6 L3VPN services over SRv6 networks

IPv6 L3VPN is an SRv6 L3 service that

- provides IPv6 Layer 3 Virtual Private Networks (VPNv6) over an SRv6 network
- enables egress Provider Edge (PE) routers to signal an SRv6 SID with the BGP overlay service route, and

- allows ingress PEs to encapsulate IPv4/IPv6 payloads in an outer IPv6 header, using the SRv6 service SID as the destination address.

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
SRv6 Services: IPv6 L3VPN	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
SRv6 Services: IPv6 L3VPN	Release 7.8.1	With this feature, the egress PE can signal an SRv6 Service SID with the BGP overlay service route. The ingress PE encapsulates the IPv4/IPv6 payload in an outer IPv6 header where the destination address is the SRv6 Service SID provided by the egress PE. BGP messages between PEs carry SRv6 Service SIDs to interconnect PEs and form VPNs.

Restrictions of SRv6-based IPv6 L3VPN

SRv6 locator assignment and allocation guidelines

- SRv6 locator can be assigned globally, for all VRFs, for an individual VRF, or per-prefix.
- Per-VRF allocation mode is supported (uDT6 behavior)
- Per-VRF-46 allocation mode is supported (uDT46 behavior)

Supported Network Capabilities

- Dual-Stack L3VPN Services (IPv4, IPv6) are supported
- Equal-Cost Multi-path (ECMP) and Unequal Cost Multipath (UCMP) are supported.
- BGP (iBGP, eBGP), OSPF, Static are supported as PE-CE protocol.
- BGP route leaking between BGP Global and L3VPN is supported.
- MPLS L3VPN and SRv6 L3VPN interworking gateway is supported.
- Per-CE allocation mode is not supported (uDX6 behavior)

Configure SRv6-based IPv6 L3VPN

To enable SRv6-based L3VPN in BGP, you must activate SRv6, specify the locator, and set the SID allocation mode. SRv6 locators can be assigned at the service, address-family, or global BGP level, with inheritance rules determining which locator is used for SID allocation.

SRv6 locators can be assigned at different levels inside the BGP routing process. BGP allocates SRv6 Service SIDs from configured locator spaces according to the following inheritance rules:

1. Use the locator as defined under the service.
If not defined under the specific service, then:
2. Use the locator as defined under the corresponding address-family.
If not defined under the corresponding address-family, then:
3. Use the locator as defined globally under BGP.

Procedure

Step 1 Enable SRv6-based L3VPN by activating SRv6 under BGP.

Specify the appropriate SRv6 locator, and configure the SID allocation mode according to your network requirements.

- Use Case 1: Assign SRv6 locator globally.

This example shows how to configure the SRv6 locator name under BGP global:

```
Node1(config)# router bgp 100
Node1(config-bgp)# segment-routing srv6
Node1(config-bgp-gbl-srv6)# locator Node1-locator
Node1(config-bgp-gbl-srv6)# exit
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::12:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv6 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust6
Node1(config-bgp-vrf)# rd 100:6
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# commit
```

- Use Case 2: Assign SRv6 locator for all VRFs.

This example shows how to configure the SRv6 locator for all VRFs under VPNv6 Address Family, with per-VRF label allocation mode:

```
Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
```

```

Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::12:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv6 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust6
Node1(config-bgp-vrf)# rd 100:6
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# commit

```

This example shows how to configure the SRv6 locator for all VRFs under VPNv4/v6 address family, with per-VRF-46 label allocation mode:

```

Node1(config)# router bgp 200
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::12:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv6 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust6
Node1(config-bgp-vrf)# rd 100:6
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# commit

```

- Use Case 3: Assign SRv6 locator for a specific VRF.

This example shows how to configure the SRv6 locator for an individual VRF, with per-VRF label allocation mode:

```

Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::12:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv6 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust6
Node1(config-bgp-vrf)# rd 100:6
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf
Node1(config-bgp-vrf-af-srv6)# commit

```


This example shows how to configure the SRv6 locator for an individual VRF, for both IPv4 and IPv6 address families, with per-VRF-46 label allocation mode:

```
Node1(config)# router bgp 200
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::12:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv6 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust6
Node1(config-bgp-vrf)# rd 100:6
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-vrf-af-srv6)# exit
Node1(config-bgp-vrf-af)# exit
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-vrf-af-srv6)# commit
```

This example shows how to configure the SRv6 locator for an Individual VRF, with per-CE label allocation mode:

```
Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::12:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv6 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust6
Node1(config-bgp-vrf)# rd 100:6
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-ce
Node1(config-bgp-vrf-af-srv6)# commit
```

Step 2 Verify that the local and received SIDs have been correctly allocated under VPNv6 and specific VRF (vrf_cust6).

Example:

```
Node1# show bgp vpnv6 unicast local-sids
```

```
BGP router identifier 10.1.1.1, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 50
BGP NSR Initial initsync version 18 (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          Local Sid          Alloc mode  Locator
Route Distinguisher: 8:8
*>i3008::8:8:8:8/128 NO SRv6 Sid      -          -
* i                NO SRv6 Sid      -          -
Route Distinguisher: 100:6 (default for vrf vrf_cust6)
*> 3001::1:1:1:1/128 fc00:0:1:40:: per-vrf    locator1
*> 3001::2:2:2:2/128 fc00:8:1:40:: per-vrf    locator2
*> 3001::3:3:3:3/128 fc00:9:1:40:: per-vrf    locator4
*> 3001::4:4:4:4/128 fc00:9:1:41:: per-ce     locator4
*> 3001::5:5:5:5/128 NO SRv6 Sid      -          -
*> 3001::12:1:1:5/128 NO SRv6 Sid      -          -
*>i3008::8:8:8:8/128 NO SRv6 Sid      -          -

```

Node1# show bgp vpnv6 unicast received-sids

```

BGP router identifier 10.1.1.1, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 50
BGP NSR Initial initsync version 18 (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          Received Sid
Route Distinguisher: 8:8
*>i3008::8:8:8:8/128 10.1.1.2          fc00:0:2:42::
* i                2400:2020:42:2fff::1    fc00:0:2:42::
                                fc00:0:2:42::
Route Distinguisher: 100:6 (default for vrf vrf_cust6)
*> 3001::1:1:1:1/128 11.1.1.2          NO SRv6 Sid
*> 3001::2:2:2:2/128 11.1.1.2          NO SRv6 Sid
*> 3001::3:3:3:3/128 11.1.1.2          NO SRv6 Sid
*> 3001::4:4:4:4/128 11.1.1.2          NO SRv6 Sid
*> 3001::5:5:5:5/128 11.1.1.2          NO SRv6 Sid
*> 3001::12:1:1:5/128 13.2.2.2          NO SRv6 Sid
*>i3008::8:8:8:8/128 10.1.1.2          fc00:0:2:42::

```

Node1# show bgp vrf vrf_cust6 local-sids

```

BGP VRF vrf_cust6, state: Active
BGP Route Distinguisher: 10.1.1.1:0
VRF ID: 0x60000004
BGP router identifier 10.1.1.1, local AS number 1
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000013 RD version: 37
BGP main routing table version 37
BGP NSR Initial initsync version 18 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          Local Sid          Alloc mode  Locator
Route Distinguisher: 8:8
*>i3008::8:8:8:8/128 NO SRv6 Sid      -          -
* i                NO SRv6 Sid      -          -
Route Distinguisher: 100:6 (default for vrf vrf_cust6)

```

```

*> 3001::1:1:1:1/128 fc00:0:1:40:: per-vrf locator1
*> 3001::2:2:2:2/128 fc00:8:1:40:: per-vrf locator2
*> 3001::3:3:3:3/128 fc00:9:1:40:: per-vrf locator4
*> 3001::4:4:4:4/128 fc00:9:1:41:: per-ce locator4
*> 3001::5:5:5:5/128 NO SRv6 Sid - -
*> 3001::12:1:1:5/128 NO SRv6 Sid - -
*>i3008::8:8:8:8/128 NO SRv6 Sid - -

```

```

Node1# show bgp vrf vrf_cust6 received-sids
BGP VRF vrf_cust6, state: Active
BGP Route Distinguisher: 100:6
VRF ID: 0x60000004
BGP router identifier 10.1.1.1, local AS number 1
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000013 RD version: 37
BGP main routing table version 37
BGP NSR Initial initsync version 18 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

```

```

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete

```

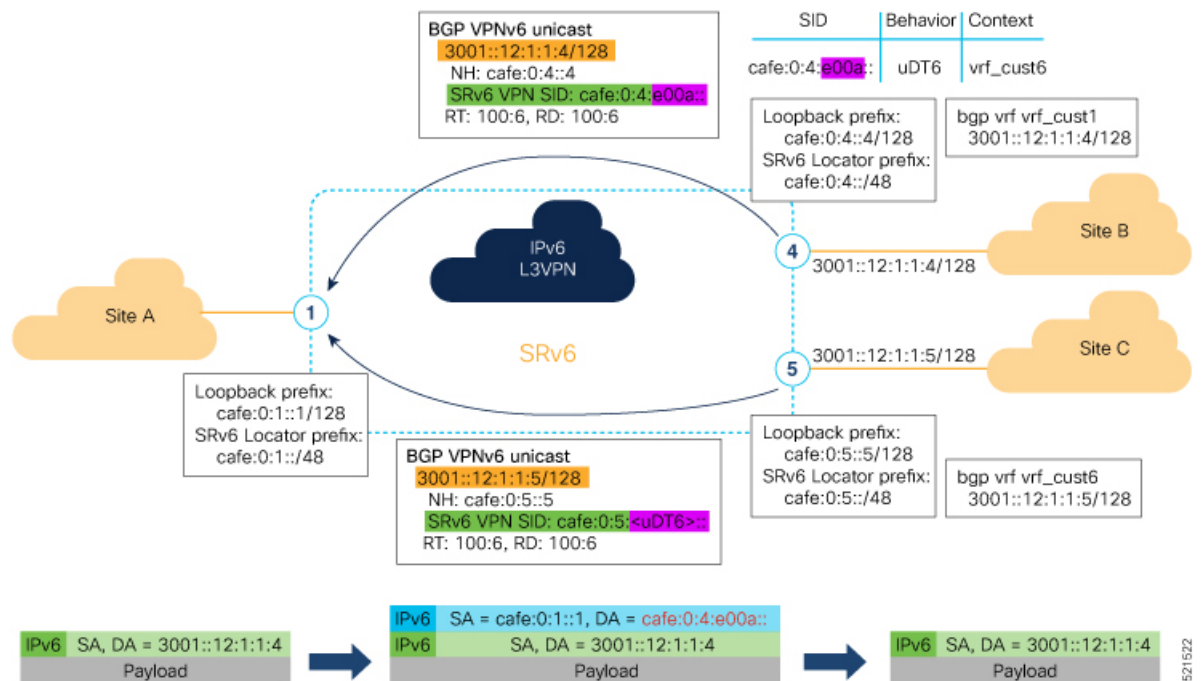
Network	Next Hop	Received Sid
Route Distinguisher: 100:6 (default for vrf vrf_cust6)		
*> 3001::1:1:1:1/128	11.1.1.2	NO SRv6 Sid
*> 3001::2:2:2:2/128	11.1.1.2	NO SRv6 Sid
*> 3001::3:3:3:3/128	11.1.1.2	NO SRv6 Sid
*> 3001::4:4:4:4/128	11.1.1.2	NO SRv6 Sid
*> 3001::5:5:5:5/128	11.1.1.2	NO SRv6 Sid
*> 3001::12:1:1:5/128	13.2.2.2	NO SRv6 Sid
*>i3008::8:8:8:8/128	10.1.1.2	fc00:0:2:42::

What to do next

[Verify SRv6-based IPv6 L3VPN configuration, on page 27](#)

Verify SRv6-based IPv6 L3VPN configuration

This example involves a VPNv6 scenario where Node1 (Ingress PE) and Node4/Node5 (Egress PEs) are configured for SRv6-based L3VPN. The provided command sequence demonstrates how to verify SRv6 L3VPN configurations for a specific VRF, using per VRF label allocation mode.



Procedure

Step 1

Observe the uDT6 SID associated with the IPv6 L3VPN, where uDT6 behavior represents Endpoint with decapsulation and IPv6 table lookup.

Example:

```
Nodel# show segment-routing srv6 sid
Fri Jan 29 19:31:53.293 UTC
```

```
*** Locator: 'Node1-locator' ***
```

SID	Behavior	Context	Owner
State RW			
-----	-----	-----	-----
---- --			
cafe:0:1:: InUse Y	uN (PSP/USD)	'default':1	sidmgr
cafe:0:1:e000:: InUse Y	uA (PSP/USD)	[Hu0/0/0/0, Link-Local]:0	isis-1
cafe:0:1:e001:: InUse Y	uA (PSP/USD)	[Hu0/0/0/1, Link-Local]:0	isis-1
cafe:0:1:e002:: InUse Y	uDT4	'vrf_cust1'	bgp-100
cafe:0:1:e003:: InUse Y	uDT4	'vrf_cust2'	bgp-100
cafe:0:1:e004:: InUse Y	uDT4	'vrf_cust3'	bgp-100
cafe:0:1:e005:: InUse Y	uDT4	'vrf_cust4'	bgp-100
cafe:0:1:e006:: InUse Y	uDT4	'vrf_cust5'	bgp-100
cafe:0:1:e007:: InUse Y	uA (PSP/USD)	[Hu0/0/0/0, Link-Local]:0:P	isis-1

```

cafe:0:1:e008::          uA (PSP/USD)      [Hu0/0/0/1, Link-Local]:0:P      isis-1
  InUse   Y
cafe:0:1:e009::          uDT6                'default'                        bgp-100
  InUse   Y
cafe:0:1:e00a::          uDT6                'vrf_cust6'                      bgp-100
  InUse   Y

```

Step 2 Verify the SRv6 based L3VPN configuration using the **show bgp vpnv6 unicast** commands on the Ingress PE.

Example:

```

Node1# show bgp vpnv6 unicast summary
Fri Jan 29 19:33:01.177 UTC
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 6
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

```

BGP is operating in STANDALONE mode.

Process Speaker	RcvTblVer	bRIB/RIB	LabelVer	ImportVer	SendTblVer	StandbyVer
	6	6	6	6	6	0

Neighbor	Spk	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	St/PfxRcd
cafe:0:4::4	0	100	122	123	6	0	0	00:20:05	1
cafe:0:5::5	0	100	111	111	0	0	0	00:49:46	1

```

Node1# show bgp vpnv6 unicast rd 100:6
Fri Jan 29 19:41:01.334 UTC
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 8
BGP NSR Initial initsync version 4 (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: 100:6 (default for vrf vrf_cust6)
*> 3001::12:1:1:1/128 ::                                0          32768 ?
*>i3001::12:1:1:4/128 cafe:0:4::4                        0          100    0 ?
*>i3001::12:1:1:5/128 cafe:0:5::5                        0          100    0 ?

```

Processed 3 prefixes, 3 paths

```

Node1# show bgp vpnv6 unicast rd 100:6 3001::12:1:1:4/128
Fri Jan 29 19:41:42.008 UTC
BGP routing table entry for 3001::12:1:1:4/128, Route Distinguisher: 100:6
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          6          6
Last Modified: Jan 29 19:29:35.858 for 00:12:06
Paths: (1 available, best #1)

```

```

Not advertised to any peer
Path #1: Received by speaker 0
Not advertised to any peer
Local, (received & used)
  cafe:0:4::4 (metric 30) from cafe:0:4::4 (1.1.1.4)
    Received Label 0xe00a00
    Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best, import-candidate,
    imported
    Received Path ID 0, Local Path ID 1, version 6
    Extended community: RT:100:6
    PSID-Type:L3, SubTLV Count:1
    SubTLV:
      T:1(Sid information), Sid:cafe:0:4::, Behavior:62, SS-TLV Count:1
      SubSubTLV:
        T:1(Sid structure):
        Source AFI: VPNv6 Unicast, Source VRF: vrf_cust6, Source Route Distinguisher: 100:6

```

Step 3 Verify the BGP prefix information for VRF instances.

Example:

```

Node1# show bgp vrf vrf_cust6 ipv6 unicast
Fri Jan 29 19:42:05.675 UTC
BGP VRF vrf_cust6, state: Active
BGP Route Distinguisher: 100:6
VRF ID: 0x60000007
BGP router identifier 1.1.1.1, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0800016 RD version: 8
BGP main routing table version 8
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop              Metric LocPrf Weight Path
Route Distinguisher: 100:6 (default for vrf vrf_cust6)
*> 3001::12:1:1:1/128 ::                      0           32768 ?
*>i3001::12:1:1:4/128 cafe:0:4::4              0         100    0 ?
*>i3001::12:1:1:5/128 cafe:0:5::5              0         100    0 ?

Processed 3 prefixes, 3 paths

Node1# show bgp vrf vrf_cust6 ipv6 unicast 3001::12:1:1:4/128

BGP routing table entry for 3001::12:1:1:4/128, Route Distinguisher: 100:6
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          17         17
Last Modified: Jan 15 16:50:44.032 for 01:48:21
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local, (received & used)
    cafe:0:4::4 (metric 30) from cafe:0:4::4 (1.1.1.4)
      Received Label 0xe00a00
      Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best, import-candidate,
      imported
      Received Path ID 0, Local Path ID 1, version 17
      Extended community: RT:100:6
      PSID-Type:L3, SubTLV Count:1
      SubTLV:

```

```

T:1(Sid information), Sid:cafe:0:4::, Behavior:62, SS-TLV Count:1
SubSubTLV:
T:1(Sid structure):
Source API: VPNv6 Unicast, Source VRF: vrf_cust6, Source Route Distinguisher: 100:6

```

Step 4 Verify the current routes in the Routing Information Base (RIB).

Example:

```

Nodel# show route vrf vrf_cust6 ipv6 unicast
Fri Jan 29 19:43:28.067 UTC

```

```

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

```

Gateway of last resort is not set

```

L   3001::12:1:1:1/128 is directly connected,
    01:01:23, Loopback105
B   3001::12:1:1:4/128
    [200/0] via cafe:0:4::4 (nexthop in vrf default), 00:13:52
B   3001::12:1:1:5/128
    [200/0] via cafe:0:5::5 (nexthop in vrf default), 00:05:53

```

```

Nodel# show route vrf vrf_cust6 ipv6 unicast 3001::12:1:1:4/128
Fri Jan 29 19:43:55.645 UTC

```

```

Routing entry for 3001::12:1:1:4/128
  Known via "bgp 100", distance 200, metric 0, type internal
  Installed Jan 29 19:29:35.696 for 00:14:20
  Routing Descriptor Blocks
    cafe:0:4::4, from cafe:0:4::4
      Nexthop in Vrf: "default", Table: "default", IPv6 Unicast, Table Id: 0xe0800000
      Route metric is 0
  No advertising protos.

```

```

Nodel# show route vrf vrf_cust6 ipv6 unicast 3001::12:1:1:4/128 detail
Fri Jan 29 19:44:17.914 UTC

```

```

Routing entry for 3001::12:1:1:4/128
  Known via "bgp 100", distance 200, metric 0, type internal
  Installed Jan 29 19:29:35.696 for 00:14:42
  Routing Descriptor Blocks
    cafe:0:4::4, from cafe:0:4::4
      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: 0x0000:100:6
      NHID:0x0(Ref:0)
      SRv6 Headend: H.Encaps.Red [f3216], SID-list {cafe:0:4:e00a::}
  Route version is 0x1 (1)
  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 3
No advertising protos.

```

Step 5 Verify the current IPv6 Cisco Express Forwarding (CEF) table.

Example:

```

Node1# show cef vrf vrf_cust6 ipv6
Fri Jan 29 19:44:56.888 UTC

```

```

::/0
  drop      default handler
3001::12:1:1:1/128
  receive   Loopback105
3001::12:1:1:4/128
  recursive  cafe:0:4::/128
3001::12:1:1:5/128
  recursive  cafe:0:5::/128
fe80::/10
  receive
ff02::/16
  receive
ff02::2/128
  receive
ff02::1:ff00:0/104
  receive
ff05::/16
  receive
ff12::/16
  receive

```

```

Node1# show cef vrf vrf_cust6 ipv6 3001::12:1:1:4/128
Fri Jan 29 19:45:23.607 UTC
3001::12:1:1:4/128, version 3, SRv6 Headend, internal 0x5000001 0x30 (ptr 0x78f2e0e0) [1], 0x0 (0x0),
0x0 (0x888a3ac8)
Updated Jan 29 19:29:35.700
Prefix Len 128, traffic index 0, precedence n/a, priority 3
  via cafe:0:4::/128, 7 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x78cd2a14 0x0]
    next hop VRF - 'default', table - 0xe0800000
    next hop cafe:0:4::/128 via cafe:0:4::/48
  SRv6 H.Encaps.Red SID-list {cafe:0:4:e00a::}

```

```

Node1# show cef vrf vrf_cust6 ipv6 3001::12:1:1:4/128 detail
Fri Jan 29 19:45:55.847 UTC
3001::12:1:1:4/128, version 3, SRv6 Headend, internal 0x5000001 0x30 (ptr 0x78f2e0e0) [1], 0x0 (0x0),
0x0 (0x888a3ac8)
Updated Jan 29 19:29:35.700
Prefix Len 128, traffic index 0, precedence n/a, priority 3
  gateway array (0x78afe238) reference count 1, flags 0x2010, source rib (7), 0 backups
    [1 type 3 flags 0x48441 (0x78ba9a60) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Jan 29 19:29:35.699
  LDI Update time Jan 29 19:29:35.701

Level 1 - Load distribution: 0
[0] via cafe:0:4::/128, recursive

  via cafe:0:4::/128, 7 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x78cd2a14 0x0]
    next hop VRF - 'default', table - 0xe0800000

```



```
next hop cafe:0:4::/128 via cafe:0:4::/48
SRv6 H.Encaps.Red SID-list {cafe:0:4:e00a::}
```

```
Load distribution: 0 1 (refcount 1)
```

Hash	OK	Interface	Address
0	Y	HundredGigE0/0/0/0	remote
1	Y	HundredGigE0/0/0/1	remote

IPv4 BGP Global SRv6 services

IPv4 BGP global is a BGP service that operates within an IPv4 network, and is supported by SRv6-based services through the implementation of uDX4, uDT4, and uDT46 SRv6 functions at the PE node.

Key concepts

- **IPv4 global BGP:** Refers to BGP routing specifically for IPv4 addresses within the global routing table, typically used for internet-wide connectivity rather than within a Virtual Routing and Forwarding (VRF) instance.
- **uDX4, uDT4, and uDT46 SRv6 Functions:** These are specific SRv6 Segment Identifiers (SIDs) or behaviors that define how a packet is processed at an SRv6-enabled node.
 - **uDX4 (Endpoint with decapsulation and IPv4 lookup):** This behavior indicates that the PE node decapsulates the SRv6 header and then performs an IPv4 lookup in its routing table to forward the packet.
 - **uDT4 (Endpoint with decapsulation and IPv4 table lookup):** This behavior is similar to uDX4, specifically indicating a lookup in an IPv4 routing table.
 - **uDT46 (Endpoint with decapsulation and IPv4/IPv6 table lookup):** This behavior supports decapsulation and a lookup in either an IPv4 or IPv6 routing table, enabling dual-stack forwarding.

Restrictions of IPv4 BGP Global SRv6 Services

- SRv6 locator can be assigned globally or under IPv4 unicast address family
- Equal-Cost Multi-path (ECMP) and Unequal Cost Multipath (UCMP) are supported.
- BGP, OSPF, Static are supported as PE-CE protocol.
- BGP route leaking between BGP Global and L3VPN is supported.
-

Configure BGP global IPv4 Over SRv6 with Per-CE SID allocation mode

Procedure

Configure BGP global IPv4 over SRv6 with per-CE SID allocation.

Example:

```
router bgp 1
  bgp router-id 10.1.0.1
  address-family ipv4 unicast
    segment-routing srv6
    alloc mode per-ce
  !
!
neighbor 60::2
  remote-as 1
  update-source Loopback1
  address-family ipv4 unicast
    route-policy passall in
    encapsulation-type srv6
    route-policy passall out
  !
!
neighbor 52.52.52.1
  remote-as 3
  address-family ipv4 unicast
    route-policy passall in
    route-policy passall out
  !
!
!
```

Configure per-VRF-46 label allocation mode

Procedure

Assign SRv6 locator

- Enable SRv6 and configure the SRv6 locator for all VRFs under VPNv4/v6 address family, with per-VRF-46 label allocation mode:

```
Node1(config)# router bgp 200
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
```

```

Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# address-family vpnv6 unicast
Node1(config-bgp-af)# vrf all
Node1(config-bgp-af-vrfall)# segment-routing srv6
Node1(config-bgp-af-vrfall-srv6)# locator Node1-locator
Node1(config-bgp-af-vrfall-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-vrfall-srv6)# exit
Node1(config-bgp-af-vrfall)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# commit

```

- Configure the SRv6 locator for an individual VRF with per-VRF label allocation mode:

```

Node1(config)# router bgp 100
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf
Node1(config-bgp-vrf-af-srv6)# commit

```

- Configure the SRv6 locator for an individual VRF for both IPv4 and IPv6 address families, with per-VRF-46 label allocation mode:

```

Node1(config)# router bgp 200
Node1(config-bgp)# address-family vpnv4 unicast
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor 3001::1:1:1:4
Node1(config-bgp-nbr)# remote-as 100
Node1(config-bgp-nbr)# address-family vpnv4 unicast
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# vrf vrf_cust1
Node1(config-bgp-vrf)# rd 100:1
Node1(config-bgp-vrf)# address-family ipv4 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6
Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-vrf-af-srv6)# exit
Node1(config-bgp-vrf-af)# exit
Node1(config-bgp-vrf)# address-family ipv6 unicast
Node1(config-bgp-vrf-af)# segment-routing srv6

```

```

Node1(config-bgp-vrf-af-srv6)# locator Node1-locator
Node1(config-bgp-vrf-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-vrf-af-srv6)# commit

```

IPv6 BGP global SRv6 services

IPv6 BGP global is a SRv6 feature that extends the support of SRv6-based BGP services to include IPv6 global BGP, and implements uDT6 and uDT46 SRv6 functions at the PE node.

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
SRv6 Services: BGP Global IPv6	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
SRv6 Services: BGP Global IPv6	Release 7.8.1	With this feature, the egress PE can signal an SRv6 Service SID with the BGP global route. The ingress PE encapsulates the IPv4/IPv6 payload in an outer IPv6 header where the destination address is the SRv6 Service SID provided by the egress PE. BGP messages between PEs carry SRv6 Service SIDs to interconnect PEs.

Key concepts

- IPv6 global BGP: Refers to BGP routing specifically for IPv6 addresses within the global routing table, typically used for internet-wide connectivity.
- uDT46 (Endpoint with decapsulation and IPv4/IPv6 table lookup): An SRv6 behavior that supports decapsulation and a lookup in either an IPv4 or IPv6 routing table, enabling dual-stack forwarding.
- SID allocation mode: Defines how SRv6 SIDs are assigned for prefixes.
 - per-VRF: Specifies that the same label is used for all routes advertised from a unique Virtual Routing and Forwarding (VRF) instance.
 - per-VRF-46: Specifies that the same service SID (uDT46 behavior) is used for all routes advertised from a unique VRF, supporting both IPv4 and IPv6.
 - per-CE (Per-Customer Edge): An allocation mode where a unique SID is allocated for each Customer Edge device.
 - route-policy: Uses a route policy to determine the SID allocation mode and locator for a given prefix.

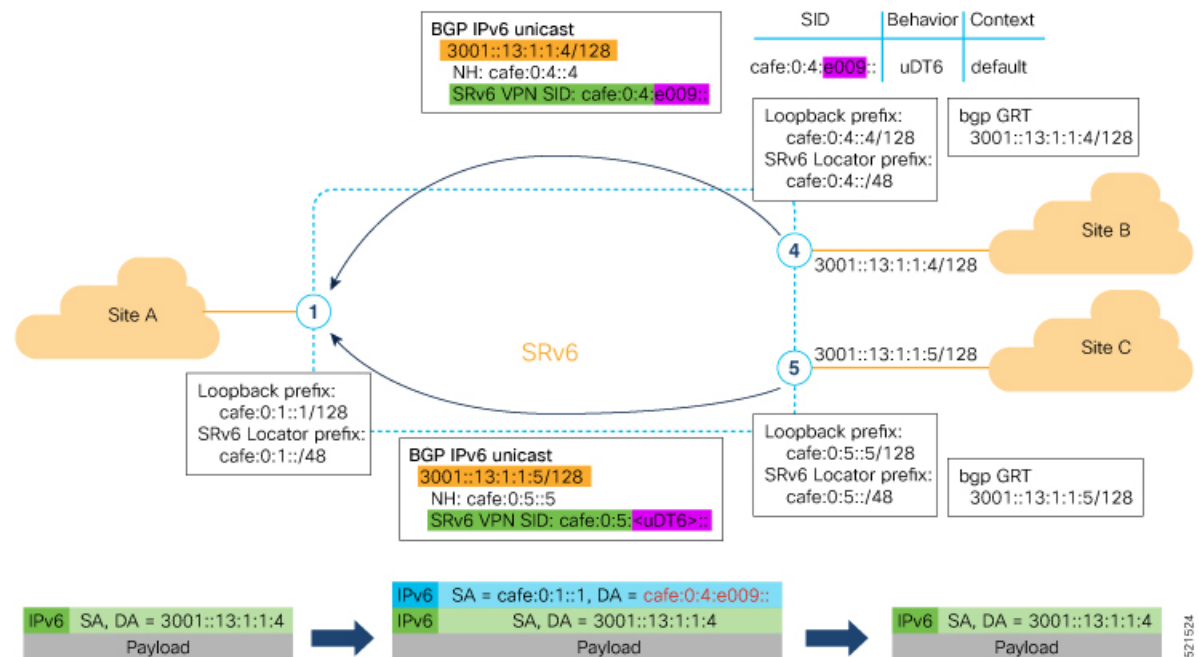
Restrictions of IPv6 BGP global SRv6 services

- SRv6 locator can be assigned globally or under IPv6 unicast address family
- Equal-Cost Multi-path (ECMP) and Unequal Cost Multipath (UCMP) are supported.
- BGP, OSPF, Static are supported as PE-CE protocol.
- Dual-Stack L3 Services (IPv4 BGP global, IPv6 BGP global) are supported.

Configure IPv6 BGP global SRv6 services

The figure shows a IPv6 BGP global scenario.

Figure 1: IPv6 BGP global scenario



Procedure

Step 1 Configure BGP global IPv6 over SRv6.

- Use Case 1: BGP Global IPv6 over SRv6 with Per-AFI SID Allocation.

The following example shows how to configure BGP global IPv6 over SRv6 with per-AFI SID allocation.

```

Node1(config)# router bgp 100
Node1(config-bgp)# bgp router-id 1.1.1.1
Node1(config-bgp)# segment-routing srv6
Node1(config-bgp-gbl-srv6)# locator Node1
Node1(config-bgp-gbl-srv6)# exit
Node1(config-bgp)# address-family ipv6 unicast
Node1(config-bgp-af)# segment-routing srv6

```

```

Node1(config-bgp-af-srv6)# locator Node1
Node1(config-bgp-af-srv6)# alloc mode per-vrf
Node1(config-bgp-af-srv6)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor cafe:0:4::4
Node1(config-bgp-nbr)# address-family ipv6 unicast
Node1(config-bgp-nbr-af)# encapsulation-type srv6
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# neighbor cafe:0:5::5
Node1(config-bgp-nbr)# address-family ipv6 unicast
Node1(config-bgp-nbr-af)# encapsulation-type srv6
Node1(config-bgp-nbr-af)# commit

```

Running Configuration

```

router bgp 100
  bgp router-id 1.1.1.1
  segment-routing srv6
    locator Node1
  !
  address-family ipv6 unicast
    segment-routing srv6
      locator Node1
      alloc mode per-vrf
    !
  !
  neighbor cafe:0:4::4
    address-family ipv6 unicast
      encapsulation-type srv6
    !
  !
  neighbor cafe:0:5::5
    address-family ipv6 unicast
      encapsulation-type srv6

```

The following example shows how to configure BGP global IPv4/IPv6 over SRv6 with uDT46 SID allocation using per-VRF-46 allocation mode (uDT46 behavior).

```

Node1(config)# router bgp 200
Node1(config-bgp)# bgp router-id 1.1.1.1
Node1(config-bgp)# segment-routing srv6
Node1(config-bgp-gbl-srv6)# locator Node1
Node1(config-bgp-gbl-srv6)# exit
Node1(config-bgp)# address-family ipv4 unicast
Node1(config-bgp-af)# segment-routing srv6
Node1(config-bgp-af-srv6)# locator Node1
Node1(config-bgp-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-srv6)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# address-family ipv6 unicast
Node1(config-bgp-af)# segment-routing srv6
Node1(config-bgp-af-srv6)# locator Node1
Node1(config-bgp-af-srv6)# alloc mode per-vrf-46
Node1(config-bgp-af-srv6)# exit
Node1(config-bgp-af)# exit
Node1(config-bgp)# neighbor cafe:0:4::4
Node1(config-bgp-nbr)# address-family ipv6 unicast
Node1(config-bgp-nbr-af)# encapsulation-type srv6
Node1(config-bgp-nbr-af)# exit
Node1(config-bgp-nbr)# exit
Node1(config-bgp)# neighbor cafe:0:5::5

```

```

Node1(config-bgp-nbr)# address-family ipv6 unicast
Node1(config-bgp-nbr-af)# encapsulation-type srv6
Node1(config-bgp-nbr-af)# commit

```

Running Configuration

```

router bgp 200
  bgp router-id 1.1.1.1
  segment-routing srv6
    locator Node1
  !
  address-family ipv4 unicast
    segment-routing srv6
    locator Node1
    alloc mode per-vrf-46
  !
  !
  address-family ipv6 unicast
    segment-routing srv6
    locator Node1
    alloc mode per-vrf-46
  !
  !
  neighbor cafe:0:4::4
    address-family ipv6 unicast
    encapsulation-type srv6
  !
  !
  neighbor cafe:0:5::5
    address-family ipv6 unicast
    encapsulation-type srv6

```

Step 2 Verify the BGP global IPv6 configuration.

The sequence of commands included correspond to router Node1 acting as Ingress PE, and routers Node4 and Node5 acting as Egress PEs.

- Verify the BGP global IPv6 configuration using the **show bgp ipv6 unicast** commands.

Example:

```

Node1# show bgp ipv6 unicast summary
Fri Jan 29 19:48:23.255 UTC
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0800000 RD version: 4
BGP main routing table version 4
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

```

BGP is operating in STANDALONE mode.

Process	RcvTblVer	bRIB/RIB	LabelVer	ImportVer	SendTblVer	StandbyVer
Speaker	4	4	4	4	4	0

Neighbor	Spk	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	St/PfxRcd
cafe:0:4::4	0	100	137	138	4	0	0	00:35:27	1
cafe:0:5::5	0	100	138	137	4	0	0	00:10:54	1

```

Node1# show bgp ipv6 unicast
Fri Jan 29 19:49:05.688 UTC
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0800000 RD version: 4
BGP main routing table version 4
BGP NSR Initial initsync version 2 (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
*> 3001::13:1:1:1/128 ::                                0           32768 i
*>i3001::13:1:1:4/128 cafe:0:4::4                0         100      0 i
*>i3001::13:1:1:5/128 cafe:0:5::5                0         100      0 i

Processed 3 prefixes, 3 paths

Node1# show bgp ipv6 unicast 3001::13:1:1:4/128
Fri Jan 29 19:49:22.067 UTC
BGP routing table entry for 3001::13:1:1:4/128
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          3         3
Last Modified: Jan 29 19:14:13.858 for 00:35:08
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    cafe:0:4::4 (metric 30) from cafe:0:4::4 (1.1.1.4)
    Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
    Received Path ID 0, Local Path ID 1, version 3
    PSID-Type:L3, SubTLV Count:1
    SubTLV:
      T:1(Sid information), Sid:cafe:0:4:e009::, Behavior:62, SS-TLV Count:1
      SubSubTLV:
        T:1(Sid structure):

```

b) Verify the current routes in the Routing Information Base (RIB):

Example:

```

Node1# show route ipv6 3001::13:1:1:4/128
Fri Jan 29 19:53:26.839 UTC

Routing entry for 3001::13:1:1:4/128
  Known via "bgp 100", distance 200, metric 0, type internal
  Installed Jan 29 19:14:13.397 for 00:35:28
  Routing Descriptor Blocks
    cafe:0:4::4, from cafe:0:4::4
      Route metric is 0
  No advertising protos.

Node1# show route ipv6 3001::13:1:1:4/128 detail
Fri Jan 29 19:50:08.601 UTC

Routing entry for 3001::13:1:1:4/128
  Known via "bgp 100", distance 200, metric 0, type internal
  Installed Jan 29 19:14:13.397 for 00:35:55

```



```

Routing Descriptor Blocks
  cafe:0:4::4, from cafe:0:4::4
    Route metric is 0
    Label: None
    Tunnel ID: None
    Binding Label: None
    Extended communities count: 0
    NHID:0x0(Ref:0)
    SRv6 Headend: H.Encaps.Red [f3216], SID-list {cafe:0:4:e009::}
Route version is 0x1 (1)
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_LOCAL
Download Priority 4, Download Version 106
No advertising protos.

```

c) Verify the current IPv6 Cisco Express Forwarding (CEF) table:

Example:

```

Node1# show cef ipv6 3001::13:1:1:4/128
Fri Jan 29 19:50:29.149 UTC
3001::13:1:1:4/128, version 106, SRv6 Headend, internal 0x5000001 0x40 (ptr 0x78 cd3944) [1],
0x0 (0x0), 0x0 (0x888a3a80)
Updated Jan 29 19:14:13.401
Prefix Len 128, traffic index 0, precedence n/a, priority 4
  via cafe:0:4::/128, 7 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x78cd2a14 0x0]
    next hop cafe:0:4::/128 via cafe:0:4::/48
    SRv6 H.Encaps.Red SID-list {cafe:0:4:e009::}

Node1# show cef ipv6 3001::13:1:1:4/128 detail
Fri Jan 29 19:51:00.920 UTC
3001::13:1:1:4/128, version 106, SRv6 Headend, internal 0x5000001 0x40 (ptr 0x78cd3944) [1], 0x0
(0x0), 0x0 (0x888a3a80)
Updated Jan 29 19:14:13.401
Prefix Len 128, traffic index 0, precedence n/a, priority 4
  gateway array (0x78afe150) reference count 1, flags 0x2010, source rib (7), 0 backups
    [1 type 3 flags 0x48441 (0x78ba99e8) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Jan 29 19:14:13.401
LDI Update time Jan 29 19:14:13.401

Level 1 - Load distribution: 0
[0] via cafe:0:4::/128, recursive

  via cafe:0:4::/128, 7 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x78cd2a14 0x0]
    next hop cafe:0:4::/128 via cafe:0:4::/48
    SRv6 H.Encaps.Red SID-list {cafe:0:4:e009::}

Load distribution: 0 1 (refcount 1)

Hash  OK  Interface  Address
0     Y   HundredGigE0/0/0/0  remote
1     Y   HundredGigE0/0/0/1  remote

```

SRv6 double recursion for multilayer BGP underlay

SRv6 double recursion is a SRv6 feature that supports

- network services, such as BGP VPNs for Layer 2 and Layer 3, that require multiple resolution layers, where one routing layer resolves over another before reaching its final destination
- three-level load balancing by collapsing the underlay (typically involving protocols like IGP or BGP in the forwarding chain), and
- specific configurations in RIB, BGP, and static routes to indicate IPv4 and IPv6 unicast routes with Layer 2 prefixes that resolve as nexthop Layer 3 prefixes.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
SRv6 double recursion for multilayer BGP underlay	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8712-MOD-M • 8011-4G24Y4H-I

Feature Name	Release Information	Feature Description
SRv6 double recursion for multilayer BGP underlay	Release 24.4.1	

Feature Name	Release Information	Feature Description
		Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: Q200, P100])(select variants only*) The feature introduces support for SRv6 double recursion, where network services such as BGP VPNs (Layer 2 and Layer 3) require multiple resolution layers. Specifically, one routing layer resolves over another before reaching the final destination. Double recursion is achieved by collapsing the underlay, typically involving protocols like IGP or BGP in the packet forwarding chain. This enables three-level load balancing and an even distribution of traffic across multiple layers of the network stack. The feature is supported on the ingress Provider Edge (PE) router. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM The feature introduces these changes: CLI: • tag-map tag <value> map forwarding-hierarchy level-2-used-as-nexthop • The show cef ipv6 and show cef ipv4 commands are enhanced to include the Layer 2 prefix information, which resolve as nexthop Layer 3 prefixes.

Feature Name	Release Information	Feature Description
		YANG Data Models: • Cisco-IOS-XR-<code>un-router-rib-cfg:router</code> (see GitHub, Yang Data Models Navigator) data model.

Key benefits of SRv6 double recursion

- Supports complex network scenarios: Enables BGP VPN services to resolve over a BGP underlay, which then resolves over IGP or directly connected routes. This accommodates new architectures in data centers and service provider networks where traditional direct IGP resolution is insufficient.
- Enhances load balancing: Allows the routing platform to distribute traffic more efficiently across all available paths and recursion levels. This improves overall network performance, reduces congestion, and ensures optimal packet paths in complex topologies.
- Improves network flexibility: Provides advanced routing solutions for service provider and data center networks, supporting complex use cases that extend beyond single-layer limitations.

Single and double recursion in BGP VPN services

Understanding the resolution processes of BGP VPN services is crucial for optimizing network routing efficiency.

Table 8: Comparison of Single and Double Recursion for BGP VPN Services

Type of Recursion	Resolution Path	Description
Single Recursion	BGP VPN service → IGP reachability	The BGP VPN service directly resolves to IGP reachability. This is applicable when a BGP VPN service can resolve directly to an IGP route without any intermediary steps.
Double Recursion	BGP VPN service → BGP underlay reachability → IGP reachability.	The BGP VPN service first resolves over a BGP underlay, which then leads to IGP reachability. This is necessary when a BGP VPN service cannot resolve directly to an IGP route but must traverse a BGP underlay first. This layered resolution process effectively handles scenarios that standard single-level load balancing cannot.

Usage guidelines for SRv6 double recursion

- Configure the locator prefix for IPv6 prefixes that are Layer 2 prefixes.
- Do not use a combination of BGP-SU (BGP Service Unicast) and BGP-IP paths at Layer 2.
- For a collapsed chain, there must be an encapsulation ID on BGP if IGP is SRv6.
- If the collapsed BGP paths are a combination of IGP IPv6 and SRv6, the router filters out only the IPv6 paths.

Configure SRv6 double recursion for multilayer BGP underlay

Before you begin

- For Layer 2 IPv6 prefixes, set the locator prefix using the **prefix-set** command in RIB.
- Configure the set of IPv4 and IPv6 Layer 2 prefixes that resolve as nexthop Layer 3 prefixes using the **prefix-set** command.

Procedure

- Step 1** Enable the hardware support for BGP-LU to allow the ingress PE router to advertise and forward the MPLS-labelled unicast routes.

Example:

```
Router#config
Router(config)#hw-module profile cef bgplu enable
```

- Step 2** Configure the IPv4 and IPv6 unicast routes in the RIB, ensuring that these Layer 2 prefixes could be used as nexthop for Layer 3 prefixes.

Example:

```
Router#config
Router(config)#router bgp 100
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#table-policy level2-ipv4-policy
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#table-policy level2-ipv6-policy
Router(config-bgp-af)#commit
```

- Step 3** Assign the required tag to the IPv4 and IPv6 unicast routes.

For these IPv4 and IPv6 routes, the RIB assigns the tags and forwards the route information to the Forwarding Information Base (FIB), indicating that these Layer 2 prefixes resolve as nexthop for Layer 3 prefixes.

Example:

IPv4 Layer 2 prefixes:

```
Router(config)#route-policy level2-ipv4-policy
Router(config-rpl)#if destination in level2_prefixes-ipv4 then
Router(config-rpl-if)#set tag 100
Router(config-rpl-if)#else
```

```
Router(config-rpl-else)#pass
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#commit
```

Example:

IPv6 Layer 2 prefixes:

```
Router(config)#route-policy level2-ipv6-policy
Router(config-rpl)#if destination in level2_prefixes-ipv6 then
Router(config-rpl-if)#set tag 100
Router(config-rpl-if)#endif
Router(config-rpl)#if destination in level2_locators then
Router(config-rpl-if)#set locator-prefix
Router(config-rpl-if)#else
Router(config-rpl-else)#pass
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
```

Step 4 Map the tags in the RIB for the IPv4 and IPv6 unicast routes.

In the following example, the RIB maps all the IPv4 and IPv6 routes tagged with the value 100, which indicates that these routes resolve as nexthop for Layer 3 prefixes. The RIB adds the *FIB_UPDATE_ROUTE_FLAG_EXTN_LVL2_HAS_DEPENDENT* flag when it sends the route update to the FIB.

Example:

```
Router#config
Router(config)#router rib
Router(config-rib)#tag-map tag 100 map forwarding-hierarchy level-2-used-as-nexthop
Router(config-rib)#commit
```

Step 5 Verify the running configuration using the **show running-config** command.

Example:

```
hw-module profile cef bgplu enable
!
router bgp 100
 address-family ipv4 unicast
   table-policy level2-ipv4-policy
 !
 address-family ipv6 unicast
   table-policy level2-ipv6-policy
 !
!
router rib
 tag-map tag 100 map forwarding-hierarchy level-2-used-as-nexthop
 !
!
```

Step 6 Use the **show cef ipv4** and **show cef ipv6** commands to verify the Layer 2 collapsed prefixes.

In the following example, the IPv6 Layer 2 prefixes, which resolve as nexthop for Layer 3 prefixes are collapsed. This is indicated by the **collapsed** keyword in the output. The SRv6 SID lists indicate the different encapsulation layers or hierarchy.

Example:

```
Router#show cef ipv6 2001:DB8:A:B::1/64
Thu Jun  6 12:48:52.399 EDT
2001:DB8:A:B::1/64, version 8, SRv6 Headend, internal 0x1000001 0x0 (ptr 0x63851c98) [1], 0x1400
(0x63851da0), 0x0 (0x638b2128)
Updated Jun  6 12:41:10.589
```

Configure SRv6 double recursion for multilayer BGP underlay

```

Prefix Len 64, traffic index 0, precedence n/a, priority 0, encap-id 0x11deadbeef
gateway array (0x61e1a798) reference count 1, flags 0x10, source rib (7), 0 backups
[2 type 3 flags 0x40008501 (0x63853e38) ext 0x0 (0x0) (collapsed)]
LW-LDI[type=3, refc=1, ptr=0x63851da0, sh-ldi=0x63853e38]
gateway array update type-time 1 Jun 6 12:41:10.589
LDI Update time Jun 6 12:41:10.629
LW-LDI-TS Jun 6 12:41:10.629
Accounting: Disabled
via 2001:DB8::1/128, 1 dependency, recursive [flags 0x30000000]
path-idx 0 NHID 0x0 [0x63a2c098 0x0]
next hop 2001:DB8::1/128 via 2001:DB8::1
SRv6 H.Encaps.Red SID-list {2001:DB8:1:e002::}
SRv6 H.Insert.Red SID-list {}
SRv6 H.Insert.Red SID-list {bbbb:bbbb:3:: bbbb:bbbb:4::}
via 2001:DB8::1/128, 1 dependency, recursive [flags 0x30000000]
path-idx 1 NHID 0x0 [0x63a2c270 0x0]
next hop 2001:DB8::1/128 via 2001:DB8::1
SRv6 H.Encaps.Red SID-list {bbbb:bbbb:2:e002::}
SRv6 H.Insert.Red SID-list {}

Load distribution: 0 1 2 2 (refcount 2)

Hash OK Interface Address
0 Y UNKNOWN intf 0x00000013 10::2
1 Y UNKNOWN intf 0x00000014 20::2
2 Y UNKNOWN intf 0x00000013 10::2
3 Y UNKNOWN intf 0x00000013 10::2

```

In the following example, the IPv4 Layer 2 prefixes, which resolve as nexthop for Layer 3 prefixes are collapsed.

Example:

```

Router#show cef ipv4 209.165.201.1 detail
Output received:
Mon Dec 2 08:31:43.765 UTC
209.165.201.1/27, version 47031, internal 0x5000001 0x40 (ptr 0x98246ad8) [1], 0x0 (0x0), 0x0 (0x0)
Updated Dec 2 08:27:35.523
Prefix Len 32, traffic index 0, precedence n/a, priority 4
gateway array (0x98099098) reference count 1, flags 0x2010, source rib (7), 0 backups
[1 type 3 flags 0x40441 (0x98134438) ext 0x0 (0x0)]
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
gateway array update type-time 1 Dec 2 08:27:35.523
LDI Update time Dec 2 08:31:14.951

Level 1 - Load distribution: 0 1
[0] via 209.165.200.225/27, recursive
[1] via 209.165.200.226/27, recursive

via 209.165.200.225/27, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 0 NHID 0x0 [0x982355c8 0x0]
next hop 209.165.200.225/27 via 209.165.200.225/27

Load distribution: 0 1 2 2 (refcount 1)

Hash OK Interface Address
0 Y HundredGigE0/0/0/0/2 fe80::2
1 Y Bundle-Ether1201 fe80::2
2 Y Bundle-Ether1301 fe80::3
3 Y Bundle-Ether1301 fe80::3

via 209.165.200.226/27, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 1 NHID 0x0 [0x98235678 0x0]
next hop 209.165.200.226/27 via 209.165.200.226/27

```



```
Load distribution: 0 1 2 2 (refcount 1)
```

Hash	OK	Interface	Address
4	Y	HundredGigE0/0/0/0/2	fe80::2
5	Y	Bundle-Ether1201	fe80::2
6	Y	Bundle-Ether1301	fe80::3
7	Y	Bundle-Ether1301	fe80::3

Step 7 For IPv6 or IPv4 static routes, use the **router static** command to configure and map the tags in RIB:

Example:

```
Router#config
Router(config)#router static
Router(config-static)#address-family ipv6 unicast
Router(config-static-afi)#2001:DB8:8::/48 4::4 tag 100
Router(config-static-afi)#commit
```

Step 8 Use the **show route** command to view the configuration for IPv4 or IPv6 static routes.

Example:

The following show output command displays the IPv4 static route configuration.

```
Router#show route 209.165.201.30

Tue Dec 3 18:22:39.579 UTC

Routing entry for 209.165.201.30/27
Known via "bgp 100", distance 200, metric 0, lvl2 has dependent
Tag 100, type internal
Installed Dec 3 18:22:35.820 for 00:00:03
Routing Descriptor Blocks
1::2, from 1::2, BGP multi path
Nexthop in Vrf: "default", Table: "default", IPv6 Unicast, Table Id: 0xe0800000
Route metric is 0
1::3, from 1::3, BGP multi path
Nexthop in Vrf: "default", Table: "default", IPv6 Unicast, Table Id: 0xe0800000
Route metric is 0
No advertising protos.
```

Example:

The following show output command displays the IPv6 static route configuration.

```
Router#show route ipv6 2001:DB8:A:B::1 detail
Tue Dec 3 18:23:55.390 UTC

Routing entry for 2001:DB8:A:B::1/64
Known via "bgp 100", distance 200, metric 0, lvl2 has dependent
Tag 100, type internal
Installed Dec 3 18:22:45.835 for 00:01:09
Routing Descriptor Blocks
1::2, from 1::2
Route metric is 0
Label: None
Tunnel ID: None
Binding Label: None
Extended communities count: 0
NHID: 0x0 (Ref: 0)
Path Grouping ID: 100
SRv6 Headend: H.Encaps.Red [f3216], SID-list {fccc:bb01:2:e002::}
Route version is 0x1c (28)
No local label
IP Precedence: Not Set
QoS Group ID: Not Set
```

Configure SRv6 double recursion for multilayer BGP underlay

```
Flow-tag: Not Set
Fwd-class: Not Set
Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_LOCAL
Download Priority 4, Download Version 17265
No advertising protos.
```
