Configuring Segment Routing

This chapter contains information on how to configure segment routing.

About Segment Routing

Segment routing is a technique by which the path followed by a packet is encoded in the packet itself, similar to source routing. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with a segment routing header. Each segment is identified by a segment ID (SID) consisting of a flat unsigned 32-bit integer.

Border Gateway Protocol (BGP) segments, a subclass of segments, identify a BGP forwarding instruction. There are two groups of BGP segments: prefix segments and adjacency segments. Prefix segments steer packets along the shortest path to the destination, using all available equal-cost multi-path (ECMP) paths.

Adjacency segments steer packets onto a specific link to a neighbor.

The segment routing architecture is applied directly to the MPLS data plane.

Segment Routing Application Module

Segment Routing Application (SR-APP) module is used to configure the segment routing functionality. Segment Routing Application (SR-APP) is a separate internal process that handles all the CLIs related to segment routing. It is responsible for reserving the SRGB range and for notifying the clients about it. It is also responsible for maintaining the prefix to SID mappings. The SR-APP support is also available for the BGP, IS-IS, and OSPF protocols.

The SR-APP module maintains the following information:

- Segment routing operation state
- Segment routing global block label ranges
- Prefix SID mappings

For more information, see Configuring Segment Routing, on page 6.

NetFlow for MPLS

NetFlow identifies packet flows for ingress IP packets and provides statistics that are based on these packet flows. NetFlow does not require any change to either the packets themselves or to any networking device. You can export the data that NetFlow gathers for your flow by using a flow exporter and export this data to a remote NetFlow Collector, such as Cisco Stealthwatch. Cisco NX-OS exports flow as part of a NetFlow export User Datagram Protocol (UDP) datagram. You can export the data that NetFlow gathers for your flow by using a flow exporter and export this data to a remote NetFlow Collector, such as Cisco Stealthwatch. Cisco NX-OS exports a flow as part of a NetFlow export User Datagram Protocol (UDP) datagram.

Beginning with Cisco NX-OS Release 9.3(1), NetFlow Collector over segment routing is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9500-EX, and 9500-FX platform switches.

Beginning with Cisco NX-OS Release 9.3(5), NetFlow Collector over segment routing is supported on Cisco Nexus 9300-FX3 platform switches.

NetFlow is not supported on Cisco Nexus 9300-GX platform switches..

NetFlow Collector supports both, single and double MPLS labels. Both, default and the non-default VRF in the exporter destination configurations is supported. NetFlow does not support an MPLS data path.

Since segment routing does not support a single label, you must configure the **address-family ipv4 labeled-unicast** command under BGP neighbor and the **allocate-label** command under the bgp configuration.

sFlow Collector

Sampled flow (sFlow) allows you to monitor real-time traffic in data networks that contain switches and routers. It uses the sampling mechanism in the sFlow agent software on switches and routers to monitor traffic and to forward the sample data to the central data collector.

Beginning with Cisco NX-OS Release 9.3(1), sFlow collector over segment routing is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9500-EX, and 9500-FX platform switches.

Beginning Cisco NX-OS Release 9.3(5), sFlow collector over segment routing is supported on Cisco Nexus 9300-FX3 platform switches.

sFlow is not supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

For information on configuring sFlow, see the *Configuring sFlow* section in the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.3(x)*.

Guidelines and Limitations for Segment Routing

Segment routing has the following guidelines and limitations:

- MPLS segment routing is not supported for FEX modules.
- Beginning with Cisco NX-OS Release 9.3(1), the **segment-routing mpls** command has changed to **segment-routing**.
- When you enable MPLS segment routing on Cisco Nexus 9504 and 9508 platform switches with a -R series line card, there can be instances of the BFD sessions going down and coming back. BGP peerings, if configured with BFD, also go down and come back up. When a BGP session goes down, it withdraws routes from the hardware. This results in packet loss until the BGP session is re-established and routes are re-installed. However, once the BFD comes up, no additional flaps occurs.
- You can run segment routing under IGP(like OSPF) or by AF labeled unicast in BGP.
- Segment Routing is supported on Cisco Nexus 9300-FX platform switches and the Cisco Nexus N9K-X9736C-FX line cards.
- Segment routing and SR-EVPN are supported on Cisco Nexus C31108PC-V, C31108TC-V, and C3132Q-V switches.
- Beginning with Cisco NX-OS Release 9.3(3), you can configure Layer 3 VPNs on Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 9.3(3), segment routing and SR-EVPN is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX platform switches.
- Beginning with Cisco NX-OS Release 9.3(3), adjacency SIDs on OSPF are supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX platform switches.
- Beginning with Cisco NX-OS Release 9.3(3), segment routing with OSPF, IS-IS underlay, and BGP labeled unicast is supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX platform switches.
- BGP allocates the SRGB label for iBGP route-reflector clients only when next-hop-self is in effect (for example, the prefix is advertised with the next hop being one of the local IP/IPv6 addresses on RR). When you have configured next-hop-self on an RR, the next hop is changed for the routes that are being affected (subject to route-map filtering).
- A nondisruptive ISSU is not supported with MPLS features for Cisco Nexus 9300-EX and 9300-FX platform switches.
- Static MPLS, MPLS segment routing, and MPLS stripping cannot be enabled at the same time.
- Beginning with Cisco NX-OS Release 9.3(5), MPLS stripping is supported on Cisco Nexus 9300-GX platform switches and the following guidelines are applicable:

  - For the MPLS strip feature to work, both the **mpls strip** and the **hardware acl tap-agg** commands should be configured after the switches are reloaded.
  - When the MPLS strip is enabled on the Cisco Nexus 9300-GX platform switches, the ACL log process is not displayed.
  - MPLS strip with dot1q VLAN is not supported.
  - For all double VLAN tags, the second VLAN range should be between 2-510.
  - MPLS strip with dot1q is not supported.
  - For PACL redirect support, you must use the **mode tap-aggregation** command on the ingress TAP interface.

- Because static MPLS, MPLS segment routing, and MPLS stripping are mutually exclusive, the only segment routing underlay for multi-hop BGP is single-hop BGP. iBGP multi-hop topologies with eBGP running as an overlay are not supported.
- MPLS pop followed by a forward to a specific interface is not supported. The penultimate hop pop (PHP) is avoided by installing the Explicit NULL label as the outlabel in the label FIB (LFIB) even when the control plane installs an IPv4 Implicit NULL label.
- BGP labeled unicast and BGP segment routing are not supported for IPv6 prefixes.
- BGP labeled unicast and BGP segment routing are not supported over tunnel interfaces (including GRE and VXLAN) or with vPC access interfaces.
- MTU path discovery (RFC 2923) is not supported over MPLS label switched paths (LSPs) or segment routed paths.
- For the Cisco Nexus 9200 Series switches, adjacency statistics are not maintained for Layer 3 or MPLS adjacencies.
- For the Cisco Nexus 9500 Series switches, MPLS LSPs and segment routed paths are not supported on subinterfaces (either port channels or normal Layer 3 ports).
- For the Cisco Nexus 9500 platform switches, segment routing is supported only in the nonhierarchical routing mode.
- The BGP configuration commands **neighbor-down fib-accelerate** and **suppress-fib-pending** are not supported for MPLS prefixes.
- The uniform model as defined in RFC 2973 and RFC 3270 is not supported. Therefore, the IP DSCP bits are not copied into the imposed MPLS header.
- Reconfiguration of the segment routing global block (SRGB) results in an automatic restart of the BGP process to update the existing URIB and ULIB entries. Traffic loss occurs for a few seconds, so you should not reconfigure the SRGB in production.
- If the segment routing global block (SRGB) is set to a range but the route-map label-index delta value is outside of the configured range, the allocated label is dynamically generated. For example, if the SRGB is set to range of 16000-23999 when a route-map label-index is set to 9000, the label is dynamically allocated.
- For network scalability, Cisco recommends using a hierarchical routing design with multi-hop BGP for advertising the attached prefixes from a top-of-rack (ToR) or border leaf switch.

- BGP sessions are not supported over MPLS LSPs or segment routed paths.
- The Layer 3 forwarding consistency checker is not supported for MPLS routes.
- You can configure segment routing traffic engineering with on-demand next hop on Cisco Nexus 9000 Series switches.
- Layer 3 VPN and Layer 3 EVPN stitching for segment routing is supported on Cisco Nexus 9000 Series switches.
- Beginning with Cisco NX-OS Release 9.3(3), Layer 3 VPN and Layer 3 EVPN stitching for segment routing is supported on 9300-GX platform switches.
- You can configure OSPFv2 as an IGP control plane for segment routing on Cisco Nexus 9000 Series switches.
- Layer 3 VPN and Layer 3 EVPN Stitching for segment routing is not supported on Cisco Nexus 9364C, 9200, 9300-EX, and 9500 platform switches with the -EX line cards.
- The OSPF segment routing command and segment-routing traffic engineering with on-demand next hop is not supported on Cisco Nexus 9364C switches.
- Segment Routing is supported on Cisco Nexus 9300-FX2 and 9300-FX3 platform switches.
- Layer 3 VPN and Layer 3 EVPN Stitching for Segment Routing, the OSPF segment routing command, and the segment-routing traffic engineering with on-demand next hop is supported on Cisco Nexus 9364C switches.
- Layer 3 VPN over Segment Routing is supported on Cisco Nexus 3100, 3200, 9200, 9300, 9300-EX/FX/FX2/FX3 platform switches and Cisco Nexus 9500 platform switches with -EX/FX and -R line cards.
- Deleting the segment routing configuration removes all the related segment routing configurations including the MPLS and the traffic engineering configurations.
- If you downgrade the Cisco Nexus device from Cisco NX-OS Release 9.3(1) to the previous NX-OS releases by setting the boot variables and reloading the switch, all earlier configurations of the segment-routing MPLS are lost.
- Before performing an ISSD from Cisco NX-OS Release 9.3(1), you must disable the segment routing configuration. Failure to do so will result in the loss of the existing segment routing configurations.
- Segment routing MPLS adjacency statistics are collected based on the out label stack and the next hop on the intermediate nodes. However, in the PHP mode, the statistics are shown on all adjacencies because the same stack is shared on all the FECs.
- If segment routing is enabled on a switch, Q-in-Q tagging on a dot1Q tagged MPLS packet is not supported, packets egress with only the outer tag.

  For example: Consider an ingress port in access dot1q tunnel mode, with VLAN 100. Incoming MPLS traffic has a dot1Q tag of 200. Typically, the traffic should egress with an outer tag of 100, and inner tag of 200 (same as the tag of the incoming packet). However, the packet egresses with an outer tag and loses the inner tag.
- When an incoming MPLS packet is untagged and the ingress port is in access VLAN mode, packets egress without any tag, if segment routing is enabled.

- We recommend that you do not configure segment routing using BGP, OSPF, and IS-IS underlay simultaneously.
- Beginning with Cisco NX-OS Release 10.2(1q)F, SR-MPLS is supported on the N9K-C9332D-GX2B platform switches. However, SR PBR and MPLS strip dot1q features are not yet supported on GX2 switches.

Configuring Segment Routing

Configuring Segment Routing

Before you begin

Confirm that the following conditions are met before configuring segment routing.

- The **install feature-set mpls**, **feature-set mpls** and **feature mpls segment-routing** commands should be present before configuring the **segment-routing** command.
- If the global block is configured, the specified range is used. Otherwise, the default 16000 – 23999 range is used.
- BGP now uses both **set label-index** <*value*> configuration and the new **connected-prefix-sid-map** CLI. In case of a conflict, the configuration in SR-APP is preferred.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)# segment-routing`
	`switch(config-sr)# mpls`
	`switch(config-sr-mpls)#`
		Enables the MPLS segment routing functionality. The **no** form of this command disables the MPLS segment routing feature.

Step 3	**connected-prefix-sid-map**
	Example:
	`switch(config-sr-mpls)#`
	`connected-prefix-sid-map`
	`switch(config-sr-mpls)#`
		Configures the connected prefix segment identifier mappings.

Step 4	**global-block** <*min*> <*max*> **Example:**	Specifies the global block range for the segment routing bindings.

	Command or Action	Purpose
	`switch(config-sr-mpls)# global-block <min> <max>` `switch(config-sr-mpls)#`	

Step 5	**connected-prefix-sid-map**
	Example:
	`switch(config-sr-mpls)# connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfsid)#`
		Configures the connected prefix segment identifier mappings.

Step 6	**address-family ipv4** **Example:** `switch(config-sr-mpls-conn-pfsid)#address-family ipv4`	Configures the IPv4 address family.
Step 7	*<prefix>/<masklen>* [**index**|**absolute**] *<label>* **Example:** `switch(config-sr-mpls)# 2.1.1.5/32 absolute 201101`	The optional keywords **index** or **absolute** indicate whether the label value entered should be interpreted as an index into the SRGB or as an absolute value.

Example

See the following configuration examples of the show commands:

```
switch# show segment-routing mpls
Segment-Routing Global info

Service Name: segment-routing

State: Enabled

Process Id: 29123

Configured SRGB: 17000 - 24999

SRGB Allocation status: Alloc-Successful

Current SRGB: 17000 - 24999

Cleanup Interval: 60

Retry Interval: 180
```

The following CLI displays the clients that are registered with SR-APP. It lists the VRFs, for which the clients have registered interest.

```
switch# show segment-routing mpls clients
            Segment-Routing Mpls Client Info

Client: isis-1
    PIB index: 1    UUID: 0x41000118    PID: 29463    MTS SAP: 412
    TIBs registered:
        VRF: default Table: base
```

```
Client: bgp-1
    PIB index: 2    UUID: 0x11b    PID: 18546    MTS SAP: 62252
    TIBs registered:
        VRF: default Table: base

Total Clients: 2
```

In the **show segment-routing mpls ipv4 connected-prefix-sid-map** CLI command example, SRGB indicates whether the prefix SID is within the configured SRGB. The **Indx** field indicates that the configured label is an index into the global block. The **Abs** field indicates that the configured label is an absolute value.

If the SRGB field displays N, it means that the configured prefix SID is not within the SRGB range and it is not provided to the SR-APP clients. Only the prefix SIDs that fall into the SRGB range are given to the SR-APP clients.

```
switch# show segment-routing mpls ipv4 connected-prefix-sid-map
            Segment-Routing Prefix-SID Mappings
Prefix-SID mappings for VRF default Table base
Prefix              SID   Type Range SRGB
13.11.2.0/24        713   Indx 1     Y
30.7.7.7/32         730   Indx 1     Y
59.3.24.0/30        759   Indx 1     Y
150.101.1.0/24      801   Indx 1     Y
150.101.1.1/32      802   Indx 1     Y
150.101.2.0/24      803   Indx 1     Y
1.1.1.1/32          16013 Abs  1     Y
```

The following CLI displays the **show running-config segment-routing** output.

```
switch# show running-config segment-routing ?

> Redirect it to a file
>> Redirect it to a file in append mode
all Show running config with defaults
| Pipe command output to filter

switch# show running-config segment-routing
switch# show running-config segment-routing

!Command: show running-config segment-routing
!Running configuration last done at: Thu Dec 12 19:39:52 2019
!Time: Thu Dec 12 20:06:07 2019

version 9.3(3) Bios:version 05.39
segment-routing
    mpls
        connected-prefix-sid-map
            address-family ipv4
                2.1.1.1/32 absolute 100100

switch#
```

Enabling MPLS on an Interface

You can enable MPLS on an interface for use with segment routing.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode for the specified interface.

Step 3	[**no**] **mpls ip forwarding** **Example:** `switch(config-if)# mpls ip forwarding`	Enables MPLS on the specified interface. The **no** form of this command disables MPLS on the specified interface.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Segment Routing Global Block

You can configure the beginning and ending MPLS labels in the segment routing global block (SRGB).

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **segment-routing** **Example:**	Enters the segment routing configuration mode and enables the default SRGB of 16000 to

	Command or Action	Purpose
	`switch(config)# segment-routing` `switch(config-sr)# mpls`	23999. The **no** form of this command unallocates that block of labels. If the configured dynamic range cannot hold the default SRGB, an error message appears, and the default SRGB will not be allocated. If desired, you can configure a different SRGB in the next step.
Step 3	[**no**] **global-block** *beginning-label ending-label* **Example:** `switch(config-sr-mpls)# global-block 16000 471804`	Specifies the MPLS label range for the SRGB. Use this command if you want to change the default SRGB label range that is configured with the **segment-routing** command. The permissive values for the beginning MPLS label and the ending MPLS label are from 16000 to 471804. The **mpls label range** command permits 16 as the minimum label, but the SRGB can start only from 16000. **Note** The minimum value for the **global-block** command starts from 16000. If you upgrading from previous releases, you should modify the SRGB so that it falls within the supported range before triggering an upgrade.
Step 4	(Optional) **show mpls label range** **Example:** `switch(config-sr-mpls)# show mpls label range`	Displays the SRGB, only if the SRGB allocation is successful.
Step 5	**show segment-routing**	Displays the configured SRGB.
Step 6	**show segment-routing mpls** **Example:** `switch(config-sr-mpls)# show segment-routing mpls`	Displays the configured SRGB.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-sr-mpls)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Label Index

You can set the label index for routes that match the **network** command. Doing so causes the BGP prefix SID to be advertised for local prefixes that are configured with a route map that includes the **set label-index**

command, provided the route map is specified in the **network** command that specifies the local prefix. (For more information on the **network** command, see the "Configuring Basic BGP" chapter in the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.)

Note Segment Routing Application (SR-APP) module is used to configure the segment routing functionality. BGP now uses both **set label-index** <*value*> configuration under route-map and the new **connected-prefix-sid-map** CLI for prefix SID configuration. In case of a conflict, the configuration in SR-APP is preferred.

Note Route-map label indexes are ignored when the route map is specified in a context other than the **network** command. Also, labels are allocated for prefixes with a route-map label index independent of whether the prefix has been configured by the **allocate-label route-map** *route-map-name* command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map SRmap`
	`switch(config-route-map)#`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	[**no**] **set label-index** *index* **Example:** `switch(config-route-map)# set label-index 10`	Sets the label index for routes that match the **network** command. The range is from 0 to 471788. By default, a label index is not added to the route.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	Required: **address-family ipv4 unicast** **Example:**	Enters global address family configuration mode for the IPv4 address family.

	Command or Action	Purpose
	`switch(config-router)# address-family ipv4 unicast` `switch(config-router-af)#`	
Step 7	**network** *ip-prefix* [**route-map** *map-name*] **Example:** `switch(config-router-af)# network 10.10.10.10/32 route-map SRmap`	Specifies a network as local to this autonomous system and adds it to the BGP routing table.
Step 8	(Optional) **show route-map** [*map-name*] **Example:** `switch(config-router-af)# show route-map`	Displays information about route maps, including the label index.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-af)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Examples for Segment Routing

The examples in this section show a common BGP prefix SID configuration between two routers.

This example shows how to advertise a BGP speaker configuration of 10.10.10.10/32 and 20.20.20.20/32 with a label index of 10 and 20, respectively. It uses the default segment routing global block (SRGB) range of 16000 to 23999.

```
hostname s1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing
  mpls
  vlan 1
segment-routing
  mpls
    connected-prefix-sid-map
    address-family ipv4
    2.1.1.1/32 absolute 100100

route-map label-index-10 permit 10
  set label-index 10
route-map label-index-20 permit 10
  set label-index 20

vrf context management
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
```

```
  ip address 10.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 10.10.10.10/32

interface loopback2
  ip address 20.20.20.20/32

line console
line vty

router bgp 1
  address-family ipv4 unicast
    network 10.10.10.10/32 route-map label-index-10
    network 20.20.20.20/32 route-map label-index-20
    allocate-label all
  neighbor 10.1.1.2 remote-as 2
    address-family ipv4 labeled-unicast
```

This example shows how to receive the configuration from a BGP speaker.

```
hostname s2
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.2/24
  ipv6 address 10:1:1::2/64
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management

interface loopback1
  ip address 2.2.2.2/32
line console

line vty

router bgp 2
  address-family ipv4 unicast
    allocate-label all
  neighbor 10.1.1.1 remote-as 1
```

```
    address-family ipv4 labeled-unicast
```

This example shows how to display the configuration from a BGP speaker. The **show** command in this example displays the prefix 10.10.10.10 with label index 10 mapping to label 16010 in the SRGB range of 16000 to 23999.

```
switch# show bgp ipv4 labeled-unicast 10.10.10.10/32

BGP routing table information for VRF default, address family IPv4 Label Unicast
BGP routing table entry for 10.10.10.10/32, version 7
Paths: (1 available, best #1)
Flags: (0x20c001a) on xmit-list, is in urib, is best urib route, is in HW, , has label
  label af: version 8, (0x100002) on xmit-list
  local label: 16010

  Advertised path-id 1, Label AF advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
  AS-Path: 1 , path sourced external to AS
    10.1.1.1 (metric 0) from 10.1.1.1 (10.10.10.10)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Prefix-SID Attribute: Length: 10
        Label Index TLV: Length 7, Flags 0x0 Label Index 10

  Path-id 1 not advertised to any peer
  Label AF advertisement
  Path-id 1 not advertised to any peer
```

This example shows how to configure egress peer engineering on a BGP speaker.

```
hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
```

```
  no switchport
  ip address 13.1.1.1/24
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown
```

The following is an example of show ip route vrf 2 command.

```
show ip route vrf 2
IP Route Table for VRF "2"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

41.11.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.9%default, [20/0], 13:26:48, bgp-2, external, tag 11 (mpls-vpn)
42.11.2.0/24, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, direct
42.11.2.1/32, ubest/mbest: 1/0, attached
    *via 42.11.2.1, Vlan2, [0/0], 13:40:52, local
```

The following is an example of **show forwarding route vrf 2** command.

```
slot  1
=======

IPv4 routes for table 2/base

------------------+----------------------------------+------------------+---------------+---------------
Prefix            | Next-hop                         | Interface        | Labels
          | Partial Install
------------------+----------------------------------+------------------+---------------+---------------
0.0.0.0/32          Drop                               Null0
127.0.0.0/8         Drop                               Null0
255.255.255.255/32  Receive                            sup-eth1
*41.11.2.0/24       27.1.31.4                          Ethernet1/3           PUSH
  30002 492529
                    27.1.32.4                          Ethernet1/21          PUSH
  30002 492529
                    27.1.33.4                          port-channel23        PUSH
  30002 492529
                    27.11.31.4                         Ethernet1/3.11        PUSH
  30002 492529
                    27.11.33.4                         port-channel23.11     PUSH
  30002 492529
                    37.1.53.4                          Ethernet1/53/1        PUSH
  29002 492529
                    37.1.54.4                          Ethernet1/54/1        PUSH
  29002 492529
                    37.2.53.4                          Ethernet1/53/2        PUSH
  29002 492529
                    37.2.54.4                          Ethernet1/54/2        PUSH
  29002 492529
                    80.211.11.1                        Vlan801               PUSH
  30002 492529
```

The following is an example of **show bgp l2vpn evpn summary** command.

```
show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 2.2.2.3, local AS number 2
BGP table version is 17370542, L2VPN EVPN config peers 4, capable peers 1
1428 network entries and 1428 paths using 268464 bytes of memory
BGP attribute entries [476/76160], BGP AS path entries [1/6]
BGP community entries [0/0], BGP clusterlist entries [0/0]
476 received paths for inbound soft reconfiguration
476 identical, 0 modified, 0 filtered received paths using 0 bytes

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.9         4    11    4637    1836 17370542    0    0 23:01:40 476
1.1.1.10        4    11       0       0        0    0    0 23:01:53 Shut (Admin)
1.1.1.11        4    11       0       0        0    0    0 23:01:52 Shut (Admin)
```

The following is an example of **show bgp l2vpn evpn** command.

```
show bgp l2vpn evpn 41.11.2.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 14.1.4.1:115
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369591
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, received and used, is best path
             Imported to 2 destination(s)
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
      Origin incomplete, MED 0, localpref 100, weight 0
      Received label 492529
      Extcommunity: RT:2:20

  Path-id 1 not advertised to any peer

Route Distinguisher: 2.2.2.3:113
BGP routing table entry for [5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224, version 17369595
Paths: (1 available, best #1)
Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: external, path is valid, is best path
             Imported from 14.1.4.1:115:[5]:[0]:[0]:[24]:[41.11.2.0]:[0.0.0.0]/224
  AS-Path: 11 , path sourced external to AS
    1.1.1.9 (metric 0) from 1.1.1.9 (14.1.4.1)
```

Configuring Segment Routing with IS-IS Protocol

About IS-IS

IS-IS is an Interior Gateway Protocol (IGP) based on Standardization (ISO)/International Engineering Consortium (IEC) 10589 and RFC 1995. Cisco NX-OS supports Internet Protocol version 4 (IPv4) and IPv6. IS-IS is a dynamic link-state routing protocol that can detect changes in the network topology and calculate loop-free routes to other nodes in the network. Each router maintains a link-state database that describes the state of the network and sends packets on every configured link to discover neighbors. IS-IS floods the link-state information across the network to each neighbor. The router also sends advertisements and updates on the link-state database through all the existing neighbors

Segment routing on the IS-IS protocol supports the following:

- IPv4
- Level 1, level 2, and multi-level routing
- Prefix SIDs
- Multiple IS-IS instances on the same loopback interface for domain border nodes
- Adjacency SIDs for adjacencies

Configuring Segment Routing with IS-IS Protocol

You can configure segment routing with IS-IS protocol.

Before you begin

IS-IS segment routing is fully enabled when the following conditions are met:

- The **mpls segment-routing** feature is enabled.
- The IS-IS feature is enabled.
- Segment routing is enabled for at least one address family under IS-IS.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	**net** *network-entity-title*	Configures the NET for this IS-IS instance.
Step 4	**address-family** *ipv4* **unicast**	Enters address family configuration mode.
Step 5	**segment-routing mpls**	Configures segment routing with IS-IS protocol.

	Command or Action	Purpose
		Note • The IS-IS command is supported only on the IPv4 address family. It is not supported on the IPv6 address family. • Redistribution is not supported from any other protocol to ISIS for the SR prefixes. You need to enable **ip router isis** command on all the prefix SID interfaces.

Configuring Segment Routing with OSPFv2 Protocol

About OSPF

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

Segment routing configuration on the OSPF protocol can be applied at the process or the area level. If you configure segment routing at the process level, it is enabled for all the areas. However, you can enable ore disable it per area level.

Segment routing on the OSPF protocol supports the following:

- OSPFv2 control plane
- Multi-area
- IPv4 prefix SIDs for host prefixes on loopback interfaces
- Adjacency SIDs for adjacencies

Adjacency SID Advertisement

OSPF supports the advertisement of segment routing adjacency SID. An Adjacency Segment Identifier (Adj-SID) represents a router adjacency in Segment Routing.

A segment routing-capable router may allocate an Adj-SID for each of its adjacencies and an Adj-SID sub-TLV is defined to carry this SID in the Extended Opaque Link LSA.

OSPF allocates the adjacency SID for each OSPF neighbor if the OSPF adjacency which are in two way or in FULL state. OSPF allocates the adjacency SID only if the segment routing is enabled. The label for adjacency SID is dynamically allocated by the system. This eliminates the chances of misconfiguration, as this has got only the local significance.

Connected Prefix-SID

OSPFv2 supports the advertisement of prefix SID for address associated with the loopback interfaces. In order to achieve this, OSPF uses Extended Prefix Sub TLV in its opaque Extended prefix LSA. When OSPF receives this LSA from its neighbor, SR label is added to the RIB corresponding to received prefix based upon the information present in extended prefix sub TLV.

For configuration, segment-routing has to be enabled under OSPF and corresponding to loopback interface that is configured with OSPF, prefix-sid mapping is required under the segment routing module.

Note SID will only be advertised for loopback addresses and only for intra-area and inter-area prefix types. No SID value will be advertised for external or NSSA prefixes.

Prefix Propagation Between Areas

To provide segment routing support across the area boundary, OSPF is required to propagate SID values between areas. When OSPF advertises the prefix reachability between areas, it checks if the SID has been advertised for the prefix. In a typical case, the SID value come from the router, which contributes to the best path to the prefix in the source area. In this case, OSPF uses such SID and advertises it between the areas. If the SID value is not advertised by the router which contributes to the best path inside the area, OSPF will use the SID value coming from any other router inside the source area.

Segment Routing Global Range Changes

OSPF advertises it's segment routing capability in terms of advertising the SID/Label Range TLV. In OSPFv2, SID/Label Range TLV is a carried in Router Information LSA.

The segment routing global range configuration will be under the “segment-routing mpls” configuration. When the OSPF process comes, it will get the global range values from segment-routing and subsequent changes should be propagated to it.

When OSPF segment routing is configured, OSPF must request an interaction with the segment routing module before OSPF segment routing operational state can be enabled. If the SRGB range is not created, OSPF will not be enabled. When an SRGB change event occurs, OSPF makes the corresponding changes in it's sub-block entries.

Conflict Handling of SID Entries

In an ideal situation, each prefix should have unique SID entries assigned.

When there is a conflict between the SID entries and the associated prefix entries use any of the following methods to resolve the conflict:

- Multiple SIDs for a single prefix - If the same prefix is advertised by multiple sources with different SIDs, OSPF will install the unlabeled path for the prefix. The OSPF takes into consideration only those SIDs that are from reachable routers and ignores those from unreachable routers. When multiple SIDs are advertised for a prefix, which is considered as a conflict, no SID will be advertised to the attached-areas for the prefix. Similar logic will be used when propagating the inter-area prefixes between the backbone and the non-backbone areas.

- Out of Range SID - For SIDs that do not fit in our SID range, labels are not used while updating the RIB.

MPLS Forwarding on an Interface

MPLS forwarding must be enabled before segment routing can use an interface. OSPF is responsible for enabling MPLS forwarding on an interface.

When segment routing is enabled for a OSPF topology, or OSPF segment routing operational state is enabled, it enables MPLS for any interface on which the OSPF topology is active. Similarly, when segment routing is disabled for a OSPF topology, it disables the MPLS forwarding on all interfaces for that topology.

MPLS forwarding is not supported on an interface which terminates at the IPIP/GRE tunnel.

Configuring Segment Routing with OSPFv2

Configure segment routing with OSPFv2 protocol.

Before you begin

Confirm that the following conditions are met before configuring segment routing with OSPFv2:

- The OSPFv2 feature is enabled.
- The segment-routing feature is enabled.
- Segment routing is enabled under OSPF.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**]**router ospf** *process* **Example:** `switch(config)# router ospf test`	Enables the OSPF mode.
Step 3	**segment-routing** **Example:** `switch(config-router)# segment-routing mpls`	Configures the segment routing functionality under OSPF.

Configuring Segment Routing on OSPF Network- Area Level

Before you begin

Before you configure segment routing on OSPF network, OSPF must be enabled on your network.

Procedure

	Command or Action	Purpose
Step 1	**router ospf process** **Example:** `switch(config)# router ospf test`	Enables the OSPF mode.
Step 2	**area** <*area id*> **segment-routing** [**mpls** | **disable**] **Example:** `switch(config-router)# area 1 segment-routing mpls`	Configures segment routing mpls mode in a specific area.
Step 3	[**no**]**area** <*area id*> **segment-routing** [**mpls** | **disable**] **Example:** `switch(config-router)#area 1 segment-routing disable`	Disables segment routing mpls mode for the specified area.
Step 4	**show ip ospf** *process* **segment-routing** **Example:** `switch(config-router)# show ip ospf test segment-routing`	Shows the output for configuring segment routing under OSPF.

Configuring Prefix-SID for OSPF

This task explains how to configure prefix segment identifier (SID) index under each interface.

Before you begin

Segment routing must be enabled on the corresponding address family.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	[**no**]**router ospf** *process* **Example:** `switch(config)# router ospf test`	Configures OSPF.
Step 3	**segment-routing** **Example:**	Configures the segment routing functionality under OSPF.

	Command or Action	Purpose

	`switch(config-router)# segment-routing`
	`switch(config-sr)#mpls`
	`switch(config-sr-mpls)#`	

Step 4	**interface loopback** *interface_number* **Example:** `switch(config-sr-mpls)# Interface loopback 0`	Specifies the interface where OSPF is enabled.
Step 5	**ip address 1.1.1.1/32** **Example:** `switch(config-sr-mpls)# ip address 1.1.1.1/32`	Specifies the IP address configured on the ospf interface.
Step 6	**ip router ospf 1 area 0** **Example:** `switch(config-sr-mpls)# ip router ospf 1 area 0`	Specifies the OSPF enabled on the interface in area.
Step 7	**segment-routing** **Example:** `switch(config-router)#segment-routing (config-sr)#mpls`	Configures prefix-sid mapping under SR module.

Step 8	**connected-prefix-sid-map**
	Example:
	`switch(config-sr-mpls)# connected-prefix-sid-map`
	`switch(config-sr-mpls-conn-pfxsid)#`
		Configures the prefix SID mapping under the segment routing module.

Step 9	**address-family ipv4**
	Example:
	`switch(config-sr-mpls-conn-pfxsid)# address-family ipv4`
	`switch(config-sr-mpls-conn-pfxsid-af)#`
		Specifies the IPv4 address family configured on the OSPF interface.

Step 10	**1.1.1.1/32 index 10** **Example:** `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 index 10`	Associates SID 10 with the address 1.1.1.1/32.
Step 11	**exit** **Example:** `switch(config-sr-mpls-conn-af)# exit`	Exits segment routing mode and returns to the configuration terminal mode.

Configuring Prefix Attribute N-flag-clear

OSPF advertises prefix SIDs via Extended Prefix TLV in its opaque LSAs. It carries flags for the prefix and one of them is N flag (Node) indicating that any traffic sent along to the prefix is destined to the router originating the LSA. This flag typically marks host routes of router's loopback.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface loopback3** **Example:** `switch(config)# interface loopback3`	Specifies the interface loopback.
Step 3	**ip ospf prefix-attributes n-flag-clear** **Example:** `switch#(config-if)# ip ospf prefix-attributes n-flag-clear`	Clears the prefix N-flag.

Configuration Examples for Prefix SID for OSPF

This example shows the configuration for prefix SID for OSPF.

```
Router ospf 10
     Segment-routing mpls
Interface loop 0
    Ip address 1.1.1.1/32
    Ip router ospf 10 area 0
Segment-routing
   Mpls
      connected-prefix-sid-m
          address-family ipv4
              1.1.1.1/32 index 10
```

Configuring Segment Routing for Traffic Engineering

About Segment Routing for Traffic Engineering

Segment routing for traffic engineering (SR-TE) takes place through a tunnel between a source and destination pair. Segment routing for traffic engineering uses the concept of source routing, where the source calculates the path and encodes it in the packet header as a segment. A Traffic Engineered (TE) tunnel is a container of TE LSPs instantiated between the tunnel ingress and the tunnel destination. A TE tunnel can instantiate one or more SR-TE LSPs that are associated with the same tunnel.

With segment routing for traffic engineering (SR-TE), the network no longer needs to maintain a per-application and per-flow state. Instead, it simply obeys the forwarding instructions provided in the packet.

SR-TE utilizes network bandwidth more effectively than traditional MPLS-TE networks by using ECMP at every segment level. It uses a single intelligent source and relieves remaining routers from the task of calculating the required path through the network.

SR-TE Policies

Segment routing for traffic engineering (SR-TE) uses a “policy” to steer traffic through the network. A SR-TE policy is a container that includes sets of segments or labels. This list of segments can be provisioned by an operator, a stateful PCE. The head-end imposes the corresponding MPLS label stack on a traffic flow to be carried over the SR-TE policy. Each transit node along the SR-TE policy path uses the incoming top label to select the next-hop, pop or swap the label, and forward the packet to the next node with the remainder of the label stack, until the packet reaches the ultimate destination.

A SR-TE policy is uniquely identified by a tuple (color, end-point). A color is represented as a 32-bit number and an end-point is an IPv4 . Every SR-TE policy has a color value. Every policy between the same node pairs requires a unique color value. Multiple SR-TE policies can be created between the same two endpoints by choosing different colors for the policies.

Cisco Nexus 9000 Series switches support the following two types of SR-TE policies:

- Dynamic SR-TE Policy - When you configure dynamic path preference under the SR-TE policy configuration or an on-demand color configuration, the path computation engine (PCE) calculates the path to the destination address. Dynamic path calculation at PCE results in a list of segments/labels that gets applied to the head-end SR-TE policy, hence the traffic gets routed through the network by hitting the segments that the SR-TE policy holds.
- Explicit SR-TE Policy - An explicit path is a list of labels, each representing a node or link in the explicit path. This feature is enabled through the **explicit-path** command that allows you to create an explicit path and enter a configuration submode for specifying the path.

SR-TE Policy Paths

A SR-TE policy path is a list of segments that specifies the path, called a segment ID (SID) list. Every SR-TE policy consists of one or more candidate paths, which can be either a dynamic or an explicit path. The SR-TE policy instantiates a single path and the selected path is the preferred valid candidate path.

You can also add on-demand color with dynamic path option and explicit policy configuration with an explicit path option for the same color and endpoint. In this case, a single policy is created on the head-end and the path with the highest preference number configured is used for forwarding traffic.

The following two methods are used to compute the SR-TE policy path:.

- Dynamic Path - When you specify the dynamic PCEP option while configuring the path preference under an on-demand color configuration or a policy configuration, the path computation is delegated to a path computation engine(PCE).
- Explicit Path - This path is an explicitly specified SID-list or a set of SID-lists.

Beginning with Cisco NX-OS Release 10.2(2)F, you can lockdown or shutdown an SR-TE policy or perform both; shutdown preference(s) of an SR-TE policy or an on-demand color template; force a specific preference to be active path option for SRTE policy; or force path re-optimization for all or a specific SRTE policy. This

feature is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX, and N9K-C9332D-GX2B platform switches. For more information, see Configuring SR-TE Manual Preference Selection, on page 36.

For more information about the Cisco Nexus 9000 switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

Affinity and Disjoint Constraints

Affinity Constraints - You can assign attributes to a link which gets advertised to path computation engine (PCE). SRTE process hosts the affinity-map and interface level configurations. Routing protocol(IGP) will register for interface updates and SRTE will notify IGP with interface updates. IGP tlvs will be passed to BGP to advertise it to external peers. There are three types of affinity constraints:

- exclude-any: specifies that links that have any of the specified affinity colors must not be traversed by the path.
- include-any: specifies that only links that have any of the specified affinity colors must be traversed by the path. Thus, links that do not have any of the specified affinity colors must not be used.
- include-all: specifies that only links that have all of the specified affinity colors must be traversed by the path. Thus, links that do not have all of the specified affinity colors must not be used.

Disjoint Constraints - You can assign disjoint constraints to the SR-TE policies which gets advertised to the PCE. The PCE then provides the disjoint path for the policies that share the same association group ID and the disjoint disjointness type.

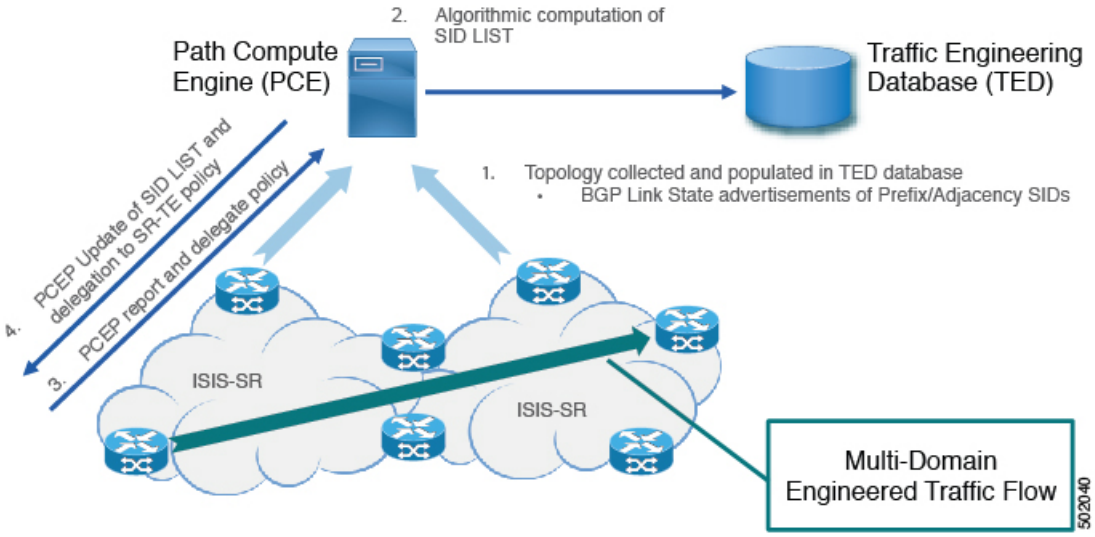
Cisco NX-OS Release 9.3(1) supports the following disjoint path levels :

- Link – The paths transit different links (but may transit same nodes).
- Node disjointness – The paths transit different links but may transit same node.

Segment Routing On Demand Next Hop

On-Demand Next hop (ODN) leverages upon BGP Dynamic SR-TE capabilities and adds the path computation (PCE) ability to find and download the end to end path based on the requirements. ODN triggers an SR-TE auto-tunnel based on the defined BGP policy. As shown in the following figure, an end-to-end path between ToR1 and AC1 can be established from both ends based on IGP Metric. The work-flow for ODN is summarized as follows:

Figure 1: ODN Operation



Guidelines and Limitations for SR-TE

SR-TE has the following guidelines and limitations:

- SR-TE ODN for both, IPv4 and IPv6 overlay is supported.
- SR-TE ODN is supported only with IS-IS underlay.
- Forwarding does not support routes with recursive next hops, where the recursive next hop resolves to a route with a binding SID.
- Forwarding does not support mixing paths with binding labels and paths without binding labels for the same route.
- The affinity and disjoint constraints are applicable only to those SR-TE policies that have a dynamic PCEP option.
- XTC supports only two policies with disjointness in the same group.
- When configuring the SR-TE affinity interfaces, the interface range is not supported.
- A preference cannot have both, the dynamic PCEP and the explicit segment lists configured together for the same preference.
- Only one preference can have a dynamic PCEP option per policy.
- For explicit policy, when configuring ECMP paths under same preference, if the first hop (NHLFE) is same for both the ECMP paths, ULIB will only install one path in switching. This occurs because both the ECMP paths create the same SRTE FEC as the NHLFE is same for both.
- In Cisco NX-OS Release 9.3(1), unprotected mode with affinity configuration is not supported by PCE (XTC).

- Beginning with Cisco NX-OS Release 9.3(3), SR-TE ODN, policies, policy paths, and the affinity and disjoint constraints are supported on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.
- Beginning with Cisco NX-OS Release 10.2(2)F, few new show commands for SR-TE policy are introduced and the autocomplete feature is provided for some of the existing SR-TE policy commands to improve usability. This feature is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX, and N9K-C9332D-GX2B platform switches.

Note For more information about the Cisco Nexus 9000 switches that support various features spanning release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

Configuring SR-TE

You can configure segment routing for traffic engineering.

Before you begin

You must ensure that the mpls segment routing feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**segment-routing**	Enters the segment-routing mode
Step 3	**traffic-engineering**	Enters the traffic engineering mode.
Step 4	**encapsulation mpls source ipv4** *tunnel_ip_address*	Configures the source address for the SR-TE Tunnel.
Step 5	**pcc**	Enters the PCC mode.
Step 6	**source-address ipv4** *pcc_source_address*	Configure source address for the PCC
Step 7	**pce-address ipv4 pce_source_address** *precedence* **num**	Configure IP address of the PCE. The lowest numbered PCE will take precedence, and the other(s) be used as a backup.
Step 8	**on-demand color** *color_num*	Enters the on-demand mode to configure the color.
Step 9	**candidate-paths**	Specifies the candidate paths of the policy.
Step 10	**preference** *preference_number*	Specifies the preference of the candidate path.
Step 11	**dynamic**	Specifies the path option.

	Command or Action	Purpose
Step 12	**pcep**	Specifies the path computation that needs to be done from the PCE.

Configuring Affinity Constraints

You can configure the affinity constraints to the SR-TE policy.

Before you begin

You must ensure that the mpls segment routing feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)# segment-routing`
	`switch(config-sr)#`
		Enables the MPLS segment routing functionality.

Step 3	**traffic-engineering**
	Example:
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		Enters the traffic engineering mode.

Step 4	**pcc**	Enters the PCC mode.
Step 5	**source-address ipv4** *pcc_source_address*	Configure source address for the PCC
Step 6	**pce-address ipv4 pce_source_address** *precedence* **num**	Configure IP address of the PCE. The lowest numbered PCE takes precedence and the other(s) are used as a backup.

Step 7	**affinity-map**
	Example:
	`switch(config-sr-te)#affinity-map`
	`switch(config-sr-te-affmap)#`
		Configures the affinity-map configuration mode.

Step 8	**color** *name* **bit-position** *position*
	Example:
	`switch(config-sr-te-affmap)# color red bit-position 2`
	`switch(config-sr-te-affmap)#`
		Configures a mapping of the user-defined name to a specific bit position in the affinity bit-map.

	Command or Action	Purpose

Step 9	**interface** *interface-name*
	Example:
	`Enter SRTE interface config mode`
	`switch(config-sr-te-if)#interface eth1/1`
	`switch(config-sr-te-if)#`
		Specifies the name of the interface. This is the affinity mapping name which refers to the specific bit in the affinity bitmap.

Step 10	**affinity**
	Example:
	`switch(config-sr-te-if)# affinity`
	`switch(config-sr-te-if-aff)#`
	`switch(config-sr-te-if-aff)# color red`
	`switch(config-sr-te-if-aff)#`
		Adds the affinity color to the interface.

Step 11	**policy** *name* | **on-demand color** *color_num*
	Example:
	`switch(config-sr-te)# on-demand color 211`
	or
	`switch(config-sr-te-color)# policy test_policy`
		Configures the policy.

Step 12	**color** *color end-point address* **Example:** `switch(config-sr-te-pol)#color 200 endpoint 2.2.2.2`	Configures the color and the end point of the policy. This is required when you are configuring the policy using the “policy name” config mode.

Step 13	**candidate-path**
	Example:
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		Specifies the candidate paths for the policy.

Step 14	**preference** *preference_number*
	Example:
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 15	**dynamic**
	Example:
	`switch(cfg-pref)# dynamic`
	`switch(cfg-dyn)#`
		Specifies the path option.

Step 16	**pcep**
	Example:
	`switch(cfg-dyn)# pcep`
	`switch(cfg-dyn)#`
		Specifies that the headend uses PCEP to request the PCE to compute a path from itself to the segment routing's policy's end point.

	Command or Action	Purpose

Step 17	**constraints**
	Example:
	`switch(cfg-dyn)# constraints`
	`switch(cfg-constraints)#`
		Enters the candidate path preference constraint mode.

Step 18	**affinity**
	Example:
	`switch(cfg-constraints)# affinity`
	`switch(cfg-const-aff)#`
		Specifies the affinity constraints of the policy.

Step 19	**exclude-any** |**include-all** | **include-any**
	Example:
	`switch(cfg-const-aff)# include-any`
	`switch(cfg-aff-inclany)#`
		Specifies the affinity constraint type. The following affinity types are available:
		• exclude-any - specifies that links that have any of the specified affinity colors must not be traversed by the path.
		• include-any - specifies that only links that have any of the specified affinity colors must be traversed by the path.
		• include-all - specifies that only links that have all of the specified affinity colors must be traversed by the path.

Step 20	**color** *color_name*
	Example:
	`switch(cfg-aff-inclany)# color blue`
	`switch(cfg-aff-inclany)#`
		Specifies the affinity color definition.

Configuring Disjoint Paths

You can configure disjoint path constraints to the SR-TE policy.

Before you begin

You must ensure that the mpls segment routing feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**segment-routing**
	Example:
	`switch(config)# segment-routing`
	`switch(config-sr)#`
		Enables the MPLS segment routing functionality.

Step 3	**traffic-engineering**
	Example:
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		Enters the traffic engineering mode.

Step 4	**pcc**	Enters the PCC mode.
Step 5	**source-address ipv4** *pcc_source_address*	Configure source address for the PCC
Step 6	**pce-address ipv4 pce_source_address** *precedence* **num**	Configure IP address of the PCE. The lowest numbered PCE takes precedence and the other(s) are used as a backup.

Step 7	**policy** *name* | **on-demand color** *color_num*
	Example:
	`switch(config-sr-te)# on-demand color 211`
	or
	`switch(config-sr-te-color)# policy test_policy`
		Configures the policy.

Step 8	**color** *color end-point address* **Example:** `switch2(config-sr-te-pol)# color 200 endpoint 2.2.2.2`	Configures the color and the end point of the policy. This is required when you are configuring the policy using the “policy name” config mode.

Step 9	**candidate-path**
	Example:
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		Specifies the candidate-paths for the policy

Step 10	**preference** *preference_number*
	Example:
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 11	**dynamic**
	Example:
	`switch(cfg-pref)# dynamic`
	`switch(cfg-dyn)#`
		Specifies the path option.

	Command or Action	Purpose

Step 12	**pcep**
	Example:
	`switch(cfg-dyn)# pcep`
	`switch(cfg-dyn)#`
		Specifies that the headend uses PCEP to request the PCE to compute a path from itself to the segment routing's policy's end point.

Step 13	**constraints**
	Example:
	`switch(cfg-dyn)# constraints`
	`switch(cfg-constraints)#`
		Enters the candidate path preference constraint mode.

Step 14	**association-group**
	Example:
	`switch(cfg-constraints)#`
	`association-group`
	`switch(cfg-assoc)#`
		Specifies the association group type.

Step 15	**disjoint**
	Example:
	`switch(cfg-assoc)# disjoint`
	`switch(cfg-disj)#`
		Specifies the path that belongs to the disjointness association group.

Step 16	**type** | **link** | **node** **Example:** `switch(config-if)#type link`	Specifies the disjointness group type.
Step 17	**id** *number* **Example:** `switch(config-if)#id 1`	Specifies the identifier of the association-group.

Configuration Examples for SR-TE

The examples in this section show affinity and disjoint configurations.

This example shows the mappings of a user defined name to an administrative group.

```
segment-routing
 traffic-eng
  affinity-map
   color green bit-position 0
   color blue bit-position 2
   color red bit-position 3
```

This example shows the affinity link colors red and green for the adjacency on eth1/1 and affinity link color green for the adjacency on eth1/2.

```
segment-routing
 traffic-eng
  interface eth1/1
   affinity
    color red
    color green
```

```
  !
  interface eth1/2
   affinity
    color green
```

This examples shows the affinity constraints for the policy.

```
segment-routing
  traffic-engineering
    affinity-map
      color blue bit-position 0
      color red  bit-position 1
    on-demand color 10
      candidate-paths
        preference 100
          dynamic
            pcep
          constraints
            affinity
              [include-any|include-all|exclude-any]
                color <col_name>
                color <col_name>
    policy new_policy
      color 201 endpoint 2.2.2.0
      candidate-paths
        preference 200
          dynamic
            pcep
          constraints
            affinity
              include-all
                color red
```

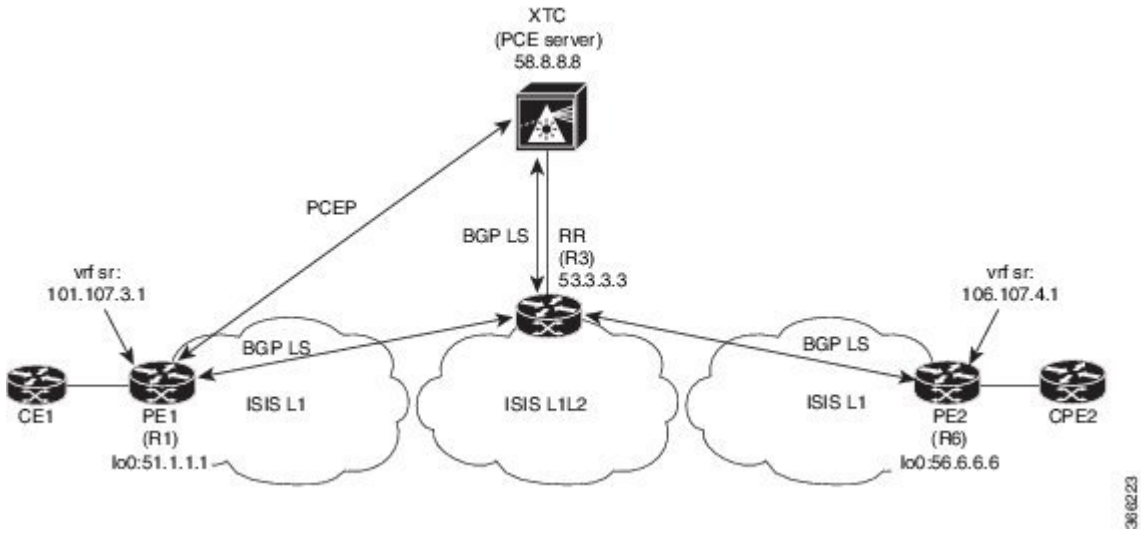
This examples shows the disjoint constraints for the policy.

```
segment-routing
 traffic-eng
  on-demand color 99
   candidate-paths
     preference 100
       dynamic
         pcep
       constraints
         association-group
           disjoint
             type link
             id 1
```

Configuration Example for an SR-TE ODN - Use Case

Perform the following steps to configure ODN for SR-TE. The following figure is used as a reference to explain the configuration steps.

Figure 2: Reference Topology



1. Configure all links with IS-IS point-to-point session from PE1 to PE2. Also, configure the domains as per the above topology.
2. Enable “distribute link-state” for IS-IS session on R1, R3, and R6.

```
router isis 1
  net 31.0000.0000.0000.712a.00
  log-adjacency-changes
  distribute link-state
  address-family ipv4 unicast
    bfd
    segment-routing mpls
    maximum-paths 32
    advertise interface loopback0
```

3. Configure the router R1 (headend) and R6 (tailend) with a VRF interface.

   VRF configuration on R1:

```
interface Ethernet1/49.101
encapsulation dot1q 201
  vrf member sr
  ip address 101.10.1.1/24
  no shutdown

vrf context sr
  rd auto
  address-family ipv4 unicast
    route-target import 101:101
    route-target import 101:101 evpn
    route-target export 101:101
    route-target export 101:101 evpn
router bgp 6500
  vrf sr
    bestpath as-path multipath-relax
    address-family ipv4 unicast
      advertise l2vpn evpn
```

4. Tags VRF prefix with BGP community on R6 (tailend).

```
route-map color1001 permit 10
  set extcommunity color 1001
```

5. Enable BGP on R6 (tailend) and R1 (headend) to advertise and receive VRF SR prefix and match on community set on R6 (tailend).

   R6 < EVPN > R3 < EVPN > R1

   BGP Configuration R6:

```
router bgp 6500
  address-family ipv4 unicast
     allocate-label all
  neighbor 53.3.3.3
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
     route-map Color1001 out
      encapsulation mpls
```

   BGP Configuration R1:

```
router bgp 6500
  address-family ipv4 unicast
     allocate-label all
  neighbor 53.3.3.3
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
       encapsulation mpls
```

6. Enable BGP configuration on R3 and BGP LS with XTC on R1, R3.abd

   BGP Configuration R3:

```
router bgp 6500
  router-id 2.20.1.2
address-family ipv4 unicast
allocate-label all
address-family l2vpn evpn
retain route-target all
  neighbor 56.6.6.6
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
       route-reflector-client
       route-map NH_UNCHANGED out
      encapsulation mpls
  neighbor 51.1.1.1
    remote-as 6500
    log-neighbor-changes
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
      route-reflector-client
      route-map NH_UNCHANGED out
      encapsulation mpls
neighbor 58.8.8.8
    remote-as 6500
```

```
    log-neighbor-changes
    update-source loopback0
    address-family link-state

route-map NH_UNCHANGED permit 10
  set ip next-hop unchanged
```

BGP Configuration R1:

```
router bgp 6500
neighbor 58.8.8.8
              remote-as 6500
               log-neighbor-changes
               update-source loopback0
               address-family link-state
```

BGP Configuration R6:

```
outer bgp 6500
   neighbor 58.8.8.8
    remote-as 6500
       log-neighbor-changes
       update-source loopback0
       address-family link-state
```

7. Enable PCE and SR-TE tunnel configurations on R1.

```
segment-routing
  traffic-engineering
    pcc
      source-address ipv4 51.1.1.1
      pce-address ipv4 58.8.8.8
    on-demand color 1001
      metric-type igp
```

Configuring SR-TE Manual Preference Selection

This section describes the configuration and execution commands introduced to support manual preference selection feature.

Guidelines and Limitations for SR-TE Manual Preference Selection

The following guidelines and limitations apply to the SR-TE manual preference selection feature:

- Beginning with Cisco NX-OS Release 10.2(2)F, the SR-TE manual preference selection feature allows you to lockdown, shutdown, or perform both on an SRTE policy or an on-demand color template; shutdown preference(s) of an SR-TE policy or an on-demand color template. Furthermore, this feature also allows you to force a specific preference to be active for the SR-TE policy and force path re-optimization for all or a specific SR-TE policy.

  This feature is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX, and N9K-C9332D-GX2B platform switches.

About SR-TE Manual Preference – Lockdown and Shutdown

Beginning with Cisco NX-OS Release 10.2(2)F, you can perform the following actions as appropriate:

- Lockdown an SRTE policy – You can enable lockdown under on-demand color templates or explicit policies. Lockdown disables auto re-optimization of path preferences for a policy. In case a new higher preferred path comes up for a policy which is locked down, then it does not automatically switch to use the new path and continues to use the current active path option until it is valid.

Note If an explicit policy configuration exists for the same color as the on-demand template, then the policy configuration takes precedence over the template configuration for the lockdown.

Example

Consider a scenario where there are multiple preferences on a policy. Assume that the higher preference path goes down due to some fault in the network. The fault could be an impending failure of a node in the higher preference path. When investigating and rectifying the fault, the operations team may need to reload or disable the problematic node and prevent any disruptions while this occurs. Then, locking down the lower preference path and preventing switching back to the higher preference path is a good option to use.

- Shutdown an SRTE policy – You can enable shutdown under on-demand color templates or explicit policies. The policy state changes to admin down, and a policy down notification is sent to all the clients interested in the policy. Disabling shutdown under on-demand color configuration changes the policy state to up or down based on the path validity of the policy.

Note If an explicit policy configuration exists for the same color as the on-demand template, then the policy configuration takes precedence over the template configuration for the shutdown.

- Shutdown preference[s] of an SRTE policy – You can shut down a path preference under an on-demand color template configuration or under a path preference of explicit policy configuration. This disables that path preference and stops it from entering any future path re-optimization until the preference is unshut. The path preference is shown as admin down or up in the output of `show srte policy` based on whether it is shut or unshut in the configuration.

Configuring SR-TE Manual Preference – Lockdown/Shutdown

You can configure lockdown, shutdown, or both on an SR-TE policy or an on-demand color template. You can also shutdown a preference under an SR-TE policy or an on-demand color template.

Before you begin

You must ensure that the mpls segment routing feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**segment-routing**	Enters the segment-routing mode.
Step 3	**traffic-engineering**	Enters the traffic engineering mode.
Step 4	**on-demand color***color_num* or **policy** *name*	Enters the on-demand mode to configure the color or configures the SR-TE policy respectively.
Step 5	(optional) [**no**] **lockdown**	Enables lockdown under the on-demand color template or explicit policy configuration. **Note** When an explicit policy configuration exists for the same color as the on-demand template, then the policy configuration takes precedence over the template configuration, and the policy is locked down.
Step 6	(optional) [**no**] **shutdown**	Shuts down any policy created from the on-demand color template or the configured SR-TE policy, as appropriate. **Note** When an explicit policy configuration exists for the same color as the on- demand template, then the policy configuration takes precedence over the template configuration, and the policy is shut down.
Step 7	**candidate-paths**	Specifies the candidate paths of the policy.
Step 8	**preference** *preference_number*	Specifies the preference of the candidate path.
Step 9	(optional) [**no**] **shutdown**	Shuts down a path preference under an SR-TE policy configuration or an on-demand color template configuration.

Force a Specific Path Preference for an SRTE Policy

To force a specific preference to be the active path option for an SRTE policy, use the `segment-routing traffic-engineering switch name <policy_name> pref <preference_number>` execution command. This command uses the preference until it is valid.

A sample output is as follows:

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
```

```
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering switch name Green_White preference 170
NX2(cfg-pref)# show srte policy Green_white detail
Policy: 8.8.8.0|801
Name: Green_White
…..
Path type = MPLS Path options count: 4
Path-option Preference:180 ECMP path count: 1 Admin: UP Forced: No
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
Path-option Preference:170 ECMP path count: 1 Admin: UP Forced: Yes Active path option
1. Explicit Weighted: No
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
```

To undo this manually selected preference, you can perform any one of the following options:

- Use the `segment-routing traffic-engineering reoptimize name <policy_name>` command. For more information, see the Force path re-optimization for an SRTE Policy or All SRTE Policies, on page 39 section.
- Switch to another preference
- Shut this policy
- Shut the selected preference

Force path re-optimization for an SRTE Policy or All SRTE Policies

When there are multiple preferences for an SRTE policy, you can re-optimize a policy, that is, pick the best preferred available path.

To force path re-optimization for a specific SRTE policy, use the `segment-routing traffic-engineering reoptimize name <policy_name>` command. The `<policy_name>` can be the name or alias name of the policy. This command undoes the preference switch command explained in the previous section and overrides lockdown if configured.

A sample output is as follows:

```
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
```

```
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:170 ECMP path count: 1
1. Explicit Weighted: Yes Weight: 1
Name: Yellow
Index: 1 Label: 16006
Index: 2 Label: 16008
NX2# segment-routing traffic-engineering reoptimize name Green_White
NX2# show srte policy Green_White
Policy: 8.8.8.0|801
Name: Green_White
Source: 2.2.2.0
End-point: 8.8.8.0
State: UP
Color: 801
Authorized: Y
Binding-sid Label: 22
Policy-Id: 3
Path type = MPLS Active path option
Path-option Preference:180 ECMP path count: 1
1. PCE Weighted: No
Delegated PCE: 11.11.11.11
Index: 1 Label: 16005
Index: 2 Label: 16008
```

To force path re-optimization for all SRTE policies, use the `segment-routing traffic-engineering reoptimize all` command to force path re-optimization for all SRTE policies present on the system. This command undoes the preference switch command explained in the previous point and overrides lockdown if configured.

Configuring SRTE Flow-based Traffic Steering

This chapter describes how to configure SRTE flow-based traffic steering on Cisco Nexus 9000-FX, 9000-FX2, 9000-FX3, 9000-GX, and 9300 platform switches.

About SRTE Flow-based Traffic Steering

The Flow-based Traffic Steering feature for Cisco NX-OS release 10.1(2) provides an alternate method of choosing the traffic to be steered, which is direct and flexible. This method allows configuring the source routing on the headend node directly, rather than on the egress node. Flow-based traffic steering allows the user to select which packets will be steered into an SRTE policy by matching fields in the incoming packet such as destination address, UDP or TCP port, DSCP bits and other properties. The matching is done by programming an ACL to steer packets into the policy.

To match and steer traffic, the Policy-Based Routing (PBR) feature is enhanced to support SRTE policies. The current PBR feature involves the RPM, ACL Manager, and AclQoS components. Beginning from Cisco NX-OS release 10.1(2), to add SRTE support, the RPM component also communicates with the SRTE and ULIB, and the communication with URIB is enhanced.

Thus, the flow-based traffic steering feature for SRTE includes the following:

- MPLS SR dataplane

- Steering IPv4 traffic is supported in default VRF, and steering IPv4 as well as IPv6 traffic is supported in non-default VRF
- Matching traffic by ACL based on a combination of the 5 tuple fields (source addr, destination addr, protocol, tcp/udp source port, tcp/udp destination port)
- Steering matched traffic into an SRTE policy
- Matching on the DSCP/TOS bits in the packet for IPv4 packets
- Matching on the Traffic Class field of the packet for IPv6 packets
- Automatic enabling and disabling of ACLs based on time-period definitions
- When steering VRF cases, support steering into an SRTE policy without specifying a next hop
- Overlay ECMP using anycast endpoints
- Packets matched by ACL take precedence over regular routes
- Flow selection based on ToS/DSCP and timer-based ACL
- The next-hop-ip is used in steering traffic to SRTE policy from one endpoint to another

Guidelines and Limitations for Flow-based Traffic Steering for SRTE

The following guidelines and limitations apply to the Flow-based Traffic Steering for SRTE feature:

- Beginning with Cisco NX-OS release 10.1(2), the flow-based traffic steering features for SRTE are supported on the Cisco Nexus 9000-FX, 9000-FX2, 9000-FX3, 9000-GX, and 9300 platform switches.
- When the SRTE policy is applied to a route-map assigned to an interface in a VRF (to steer L3VPN/L3EVPN traffic), if the next hop in the set statement resolves to a BGP prefix, and that BGP prefix is already using an SRTE policy to steer traffic, then the route-map does not steer traffic.
- Underlay ECMP is only supported if label stack is the same for each active SRTE path (ECMP member) in the policy. The 9000-GX platforms do not have this limitation.
- The route-map tracking feature is not supported.
- When steering into SRTE policies, having multiple **set next-hop** in a single route-map sequence entry is not supported.
- When the SRTE policy is applied to a route-map assigned to an interface in a VRF (to steer L3VPN/L3EVPN traffic), if the next hop in the set statement resolves to a BGP route (overlay route) that has multiple next hops in RIB, the traffic is only steered to the first next hop in the route and will not ECMP over all next hops.
- When the SRTE policy name is used in the route-map set statement, rather than color and endpoint, it can only be used for default VRF steering. If not, you must select an SRTE path that is defined explicitly. Specifically, this cannot be used to select SRTE policies defined to use a segment-list containing the policy-endpoint keyword in place of a label.
- The following keywords, which are applicable for the next hop-ip specified in the **set ip next-hop** <>, are not supported in the route-map when steering into SRTE policies:
    - verify-availability

  - drop-on-fail
  - force-order
  - load-share

- Route-map with srte-policy can be applied on the interface even if the required features (segmenting-routing, l3 evpn or l3vpn) are not enabled on the device. But the set-actions with srte-policy are kept down, that is, default-routing will be done for those flows.
- A route-map can have set commands with srte-policy and without srte-policy.
- For set-commands without srte-policy information, steering is done only if the reachability to the next-hop-ip does not require MPLS label.
- When a route-map is associated with an interface in a non-default VRF, and that route-map contains a sequence that specifies a next hop IP address **N** and an SRTE policy, then all other sequences on that route-map and all other route-maps associated with the same VRF that also use the same next hop IP address must also have an SRTE policy. Associating another route-map or route-map sequence using the same next hop IP and a different SRTE policy to the same VRF is not allowed.
- Similarly, when a route-map is associated with an interface in a non-default VRF, and that route-map does not specify an SRTE policy but specifies a next hop IP address **N**, then another sequence in that route-map or a separate route-map is not applied that uses the same next hop IP address N and specifies an SRTE policy.
- The SRTE flow-based traffic steering cannot be used at the same time as VXLAN or EoMPLS PBR.
- The SR label stats are not supported for policy based routed traffic on the SRTE ingress node. However, ACL redirect stats are supported.
- The IPv6 traffic in the default VRF cannot be steered into an SRTE policy. The MPLS SR underlay is only supported for IPv4. However, if an IPv6 SR underlay is required, use SRv6 instead.
- The 9000-FX, 9000-FX2, 9000-FX3, and 9300 platform hardware are unable to push unique underlay label stack per ECMP member, which impacts underlay ECMP on those platforms. In other words, if there are multiple active segment-lists on an SRTE policy (a single preference is configured with multiple segment-lists) where the first hop of the segment lists is different, then such a configuration is not supported. In such cases, as a workaround, configure anycast SID to make the label stack same across all ECMP members.
- Modular platforms are not supported in Cisco NX-OS release 10.1(2).
- Beginning with Cisco NX-OS release 10.2(2)F, the flow-based traffic steering features for SRTE are supported on the Cisco N9K-C9332D-GX2B platform switches.

Configuration Process: SRTE Flow-based Traffic Steering

The configuration process for the SRTE Flow-based Traffic Steering feature is as follows:

1. Configure the IP access lists, especially matching the criteria on the IP access list.

   For more information, see the *Configuring IP ACLs* chapter in the *Cisco Nexus Series NX-OS Security Configuration Guide*.

2. Define the SRTE policy.

For more information about configuring SRTE, see the *Configuring Segment Routing for Traffic Engineering* chapter in the *Cisco Nexus 9000 series NX-OS Label Switching Configuration Guide*.

3. Configure the route map that binds the match (IP access list configured in step 1) and action. The match refers to the fields to match on the packet, and the action refers to what SRTE policy to steer into and the VPN label to use, if any.

Configuring Flow Selection Based on ToS/DSCP and Timer-based ACL

In the SRTE flow-based traffic steering feature, the flow selection is based on ToS/DSCP and Timer based ACL.

Perform the following configuration procedure for the route map configuration in default and non-default VRF into a policy selected by different criteria to work properly.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**ip** | **ipv6**] **access-list** *acl_name*
	Example:
	`switch(config)# ip access-list L4_PORT`
	`switch(config)#`
		Use a name to define an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 3	**10 permit ip** *ip_address* **any**
	Example:
	`switch(config)# 10 permit ip any 5.5.0.0/16`
	`switch(config)#`
		Shows the IP or IPv6 access lists configured on the switch.

Step 4	**20 permit tcp** *tcp_address* [**any**]
	Example:
	`switch(config)# 20 permit tcp any 5.5.0.0/16`
	`switch(config)#`
		Sets TCP permit conditions for an IP or IPv6 access list.
		Note The **any** keyword is used for IPv6 only.

Step 5	[**ip** | **ipv6**] **access-list** *dscp_name*
	Example:
	`switch(config)# ip access-list dscp`
	`switch(config)#`
		Use a name to define DSCP for an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 6	**10 permit tcp any** *tcp_address* **dscp** <*dscp value*>	Set the DSCP value for an IP or IPv6 access list.

	Command or Action	Purpose

	Example:
	`switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11`
	`switch(config)#`
		Note The **any** keyword is used for IPv6 only.

Step 7	[**ip** | **ipv6**] **access-list** *acl_name*
	Example:
	`switch(config)# ip access-list acll`
	`switch(config)#`
		Use a name to define an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 8	**10 permit tcp any** *tcp_address* **acl** *acl_name*
	Example:
	`switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		Sets TCP permit conditions for an IP or IPv6 access list.
		Note The **any** keyword is used for IPv6 only.

Step 9	[**ip** | **ipv6**] **access-list** *acl_name*
	Example:
	`switch(config)# ip access-list acl1`
	`switch(config)#`
		Use a name to define an IP or IPv6 access list and enter the IP or IPv6 access-list configuration mode.

Step 10	**10 permit tcp any** *any time - range tl*
	Example:
	`switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11`
	`switch(config)#`
		Sets time range value to define the time range for TCP for an IP or IPv6 access list.
		Note The **any** keyword is used for IPv6 only.

Step 11	**time-range** *name*
	Example:
	`switch(config-acl)# time-range t1`
	`switch(config)#`
		Use a name to define the time range for an IP or IPv6 access list.

Step 12	**F2(config-time-range)#**
	WOLF2(config-time-range)#
	Example:
	`switch(config-time-range)# 10 absolute start 20:06:56 8 february 2021`
	`end 20:10:56 8 february 2021`
		Define a time range for the configuration.

Configuring Route Map in Default and Non-default VRF for Flow-based Traffic Steering

The following sections show how to configure the route map in default and non-default VRF for the SRTE flow-based traffic steering feature:

Configuring Route Map in Default VRF into a Policy Selected by Color and Endpoint

Perform the following steps to configure a route-map that steers traffic in the default VRF into a policy that is selected by color and endpoint.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy color** *num* **endpoint** *ip address*
	Example:
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		Configures the SRTE policy color and the end point of the policy.
		Note Only IPv4 address can be the endpoint.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters interface configuration mode.

Step 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Configuring Route Map in Default VRF into a Policy Selected by Name

Perform the following steps to configure a route-map that steers traffic in the default VRF into a policy that is selected by name.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose
Step 1	**route-map FLOW1** *seq_num* **Example:**	Names the route map FLOW1.

	Command or Action	Purpose
	`switch(config)# route-map FLOW1 seq 10` `switch(config-route-map)#`	

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy name** *policy-name*
	Example:
	`switch(config-route-map)# set srte-policy name policy1`
	`switch(config-route-map)#`
		Configures the SRTE policy name.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters the interface configuration mode.

Step 5	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Nexthop, Color, and Endpoint

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by color and endpoint. In this procedure, a nexthop is specified so that the correct MPLS VPN label is imposed on the traffic.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name* **Example:**	Specifies the fields the route-map should match by attaching an ACL describing the fields.

	Command or Action	Purpose
	`switch(config-route-map)# match ip address L4_PORT` `switch(config-route-map)#`	

Step 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num* **endpoint** *ip address*
	Example:
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		Redirects packet to the configured next-hop through the srte-policy (color and endpoint).

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 5	**interface** *interface-type/slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters the interface configuration mode.

Step 6	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		Adds this interface to a VRF.

Step 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 8	[**no**] **shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		Disables the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Nexthop and Color

Perform the following steps to configure a route-map that steers traffic in the default VRF into a policy that is selected by color and endpoint, but the endpoint is not explicitly configured. The nexthop is specified so that the correct MPLS VPN label is imposed on the traffic and so the correct SRTE endpoint is derived from the route matching the nexthop.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy color** *num*
	Example:
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121`
	`switch(config-route-map)#`
		Redirects packet to the configured next-hop through the srte-policy (color).

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 5	**interface** *interface-type/slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters the interface configuration mode.

Step 6	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		Adds this interface to a VRF.

Step 7	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-if)# ip policy route-map FLOW1`
	`switch(config-if-route-map)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 8	[**no**] **shutdown**
	Example:
	`switch(config-if-route-map)# no shutdown`
	`switch(config-if-route-map)#`
		Disables the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Nexthop and Name

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by name. The nexthop is specified so that the correct MPLS VPN label is imposed on the traffic

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set** [**ip** | **ipv6**] **next-hop** *destination-ip-next-hop* **srte-policy** *name*
	Example:
	`switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1`
	`switch(config-route-map)#`
		Redirects packet to the configured next-hop through the srte-policy (name).

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 5	**interface** *interface-type/slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters the interface configuration mode.

Step 6	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member vrf1`
	`switch(config-if)#`
		Adds this interface to a VRF.

Step 7	[**ip** | **ipv6**] **policy route-map FLOW1** **Example:**	Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

	Command or Action	Purpose
	`switch(config-if)# ip policy route-map FLOW1` `switch(config-if)#`	

Step 8	[**no**] **shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		Disables the interface.

Configuring Route Map in Non-default VRF into a Policy Selected by Color and Endpoint

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by color and endpoint. This procedure does not require a nexthop to be specified. The VPN label is derived by looking up the label assigned to the VRF on the local switch. This is only allowed to be configured when the same label is assigned to the VRF on all switches by using the BGP allocate-index configuration for the VRF on all switches.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy color** *num* **endpoint** *ip address*
	Example:
	`switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1`
	`switch(config-route-map)#`
		Configures the SRTE policy color and the end point of the policy.
		Note Only IPv4 address can be the endpoint.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters the interface configuration mode.

Step 5	**vrf member** *vrf-name* **Example:**	Adds this interface to a VRF.

	Command or Action	Purpose
	`switch(config-route-map-if)# vrf member vrf1` `switch(config-route-map-if)#`	

Step 6	**[ip | ipv6] policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 7	**[no] shutdown**
	Example:
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		Disables the interface.

Step 8	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 9	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enters the BGP feature.

Step 10	**router bgp as-number**
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.

Step 11	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		Associates the BGP process with a VRF.

Step 12	**allocate-index** *index* **Example:** `switch(config-router-vrf)# allocate-index 10`	Assigns an index to the VRF. This instructs BGP to allocate a static MPLS local VPN label for the VRF. The MPLS VPN label assigned to the VRF is derived from the value specified - the index is used as an offset into a special range of MPLS label values. For a given index value the same local label is always allocated.

Configuring Route Map in Non-default VRF into a Policy Selected by Name

Perform the following steps to configure a route-map that steers traffic in a non-default VRF into a policy that is selected by name. This procedure does not require a nexthop to be specified. The VPN label is derived by looking up the label assigned to the VRF on the local switch. This is only allowed to be configured when

the same label is assigned to the VRF on all switches by using the BGP allocate-index configuration for the VRF on all switches.

Before you begin

You must ensure that the MPLS segment routing traffic engineering and PBR features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**route-map FLOW1** *seq_num*
	Example:
	`switch(config)# route-map FLOW1 seq 10`
	`switch(config-route-map)#`
		Names the route map FLOW1.

Step 2	**match** [**ip** | **ipv6**] **address** *acl_name*
	Example:
	`switch(config-route-map)# match ip address L4_PORT`
	`switch(config-route-map)#`
		Specifies the fields the route-map should match by attaching an ACL describing the fields.

Step 3	**set srte-policy** *name*
	Example:
	`switch(config-route-map)# set srte-policy policy1`
	`switch(config-route-map)#`
		Configures the SRTE policy name.

Step 4	**interface** *interface-type/slot/port*
	Example:
	`switch(config-route-map)# interface ethernet 1/1`
	`switch(config-route-map-if)#`
		Enters interface configuration mode.

Step 5	**vrf member** *vrf-name*
	Example:
	`switch(config-route-map-if)# vrf member vrf1`
	`switch(config-route-map-if)#`
		Adds this interface to a VRF.

Step 6	[**ip** | **ipv6**] **policy route-map FLOW1**
	Example:
	`switch(config-route-map-if)# ip policy route-map FLOW1`
	`switch(config-route-map-if)#`
		Assigns a route map for IP or IPv6 policy-based routing to the interface. This applies the route-map for all the traffic ingressing into the interface.

Step 7	[**no**] **shutdown**
	Example:
	`switch(config-route-map-if)# no shutdown`
	`switch(config-route-map-if)#`
		Disables the interface.

	Command or Action	Purpose

Step 8	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits the route-map configuration mode and returns to the global configuration mode.

Step 9	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enters the BGP feature.

Step 10	**router bgp as-number**
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.

Step 11	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		Associates the BGP process with a VRF.

Step 12	**allocate-index** *index*
	Example:
	`switch(config-router-vrf)#`
	`allocate-index 10`
		Assigns an index to the VRF. This instructs BGP to allocate a static MPLS local VPN label for the VRF. The MPLS VPN label assigned to the VRF is derived from the value specified - the index is used as an offset into a special range of MPLS label values. For a given index value the same local label is always allocated.

Configuration Example for SRTE Flow-based Traffic Steering

This section includes the following examples for configuring flow-based traffic steering for SRTE:

Configuration Example for Flow Selection Based on ToS/DSCP and Timer-based ACL

```
switch# configure terminal
switch(config)# ip access-list L4_PORT
switch(config)# 10 permit ip any 5.5.0.0/16
switch(config)# 20 permit tcp any 5.5.0.0/16
switch(config)# ip access-list dscp
switch(config)# 10 permit tcp any 5.5.0.0/16 dscp af11
switch(config)# ip access-list acll
switch(config)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11
switch(config)# ip access-list acl1
switch(config-acl)# 10 permit tcp any 5.5.0.0/16 eq www dscp af11
switch(config-acl)# time-range t1
start 20:06:56 8 february 2021 end 20:10:56 8 february 2021
```

Configuration Example for Route Map in Default VRF into a Policy Selected by Color and Endpoint

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
```

```
switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map-if)# ip policy route-map FLOW1
```

Configuration Example for Route Map in Default VRF into a Policy Selected by Name

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy name policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map-if)# ip policy route-map FLOW1
```

Configuration Example for Route Map in Non-default VRF into a Policy Selected by Next hop, Color, and Endpoint

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

Configuration Example for Route Map in Non-default VRF into a Policy Selected by Next hop and Color

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy color 121
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

Configuration Example for Route Map in Non-default VRF into a Policy Selected by Next hop and Name

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set ip next-hop 5.5.5.5 srte-policy policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
```

Configuration Example for Route Map in Non-default VRF into a Policy Selected by Color and Endpoint

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy color 121 endpoint 10.0.0.1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
switch(config)# feature bgp
switch(config)# router bgp 1.1
switch(config-router)# vrf vrf1
switch(config-router-vrf)# allocate-index 10
```

Configuration Example for Route Map in Non-default VRF into a Policy Selected by Name

```
switch(config)# route-map FLOW1 seq 10
switch(config-route-map)# match ip address L4_PORT
switch(config-route-map)# set srte-policy policy1
switch(config-route-map)# interface ethernet 1/1
switch(config-route-map)# vrf member vrf1
switch(config-route-map-if)# ip policy route-map FLOW1
switch(config)# feature bgp
switch(config)# router bgp 1.1
switch(config-router)# vrf vrf1
switch(config-router-vrf)# allocate-index 10
```

Verifying Configuration for Flow-based Traffic Steering for SRTE

To display the appropriate details about the flow-based steering for SRTE configuration, perform one of the following tasks:

Table 1: Verifying Configuration for Flow-based Traffic Steering for SRTE

Command	Purpose
show srte policy	Displays only the authorized policies.
show srte policy [**all**]	Displays the list of all policies available in the SR-TE.
show srte policy [**detail**]	Displays the detailed view of all the requested policies.
show srte policy <*name*>	Filters the SR-TE policy with the name and displays the list of all policies available with that name in the SR-TE. **Note** This command has the autocomplete feature for the policy-name. To use this feature, add a question mark or press TAB.
show srte policy color <*color*> **endpoint** <*endpoint*>	Displays the SR-TE policy for the color and endpoint. **Note** This command has the autocomplete feature for color and endpoint. To use this feature, add a question mark or press TAB.
show route-map [**name**]	Displays information about a route map.
show forwarding mpls srte module	Displays SRTE information in Forwarding Information Base - FIB module.

Configuring MPLS OAM Monitoring for SRTE Policies

About MPLS OAM Monitoring for SRTE Policies

Beginning with Cisco NX-OS release 10.1(2), MPLS OAM monitoring allows the switch on which one or more SRTE policies are configured to proactively detect if the active path or paths of an SRTE policy have failed. If the paths in the currently active preference have all failed, SRTE will consider that preference down and so make the next highest preference on the policy active, if there is such a preference, or otherwise mark the policy as down.

Before this feature, the state of an SRTE preference and policy was only determined by the state of the first hop (the first MPLS label) of the paths in the preference. If the label was programmed the path was considered up, and if the label was missing or invalid the path was considered down.

The MPLS OAM monitoring augments this validation by sending MPLS LSPV Nil-FEC ping requests continuously along the SRTE path. Each ping request contains the same label stack as would be imposed on traffic that follows the SRTE policy, making the pings take the same path. The pings are sent with a configurable interval between each ping, and a response to the ping from the final node of the path is expected within the interval. If a failure response is returned from the final node or no response is received within the interval, it is counted as a failed interval. After a configurable number of failed intervals occur in sequence, the path is considered down. If all paths in a preference are down, then the preference is considered down.

Paths Monitored

Only when the CLIs are enabled to monitor a path using proactive monitoring will the path be monitored using OAM. Only the paths that are associated with a policy will be monitored. For example, if a segment list is created and is not associated with a policy it is not monitored. As well, if the same path is used in multiple policies only one monitoring session will be created for that path. This applies whether the path is a segment-list associated to a preference in a policy or is calculated using path completion on the headend.

By default, when the image is upgraded from a version without OAM monitoring support to a version with monitoring support, the monitoring method for policies will be the traditional first-hop method.

MPLS OAM monitoring may be enabled globally for all SRTE policies. If enabled globally, it can be selectively disabled per policy. If not enabled globally, it can be enabled selectively for individual policies.

Index Limit

The index-limit X CLI is used to ping only an initial subset of the path rather than the entire path. Only indexes in the segment list that are less than or equal to the specified index-limit are part of the path to monitor. For example, if the segment list is the following:

```
index 100 mpls label 16001
index 200 mpls label 16002
index 300 mpls label 16003
```

Then if index-limit is not specified, the path to be pinged will be 16001, 16002, 16003. If index-limit is 250, then the path to be pinged will be 16001, 16002. If index-limit is 200, then the path to be pinged will also be 16001, 16002.

Guidelines and Limitations for MPLS OAM Monitoring for SRTE Policies

The MPLS OAM monitoring for SRTE policies has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.1(2), MPLS OAM monitoring (continuous and proactive path) is introduced and supported on Cisco Nexus 9300 EX, 9300-FX, 9300-FX2, and 9300-GX platform switches.
- On the head-end node where the SRTE policies are configured, both SRTE and MPLS OAM must be separately enabled as part of feature mpls segment-routing traffic-engineering and feature mpls oam respectively. If not, the user cannot configure the monitoring of SRTE policies using OAM. In addition, the remaining nodes in the SR fabric must have MPLS OAM enabled using feature mpls oam to respond to the pings sent by MPLS OAM monitoring.
- SRTE limits the maximum number of monitoring sessions to 1000.
- The minimum interval between pings is 1000 milliseconds.
- When SRTE OAM monitoring policies are running on a device, `feature mpls oam` cannot be disabled. Only when all the SRTE OAM monitoring policies are disabled, the `feature mpls oam` can be disabled from the device. Otherwise, the following error message is displayed:

  "SRTE MPLS liveness detection is either enabled for all policies, is enabled for at least one policy, or is enabled for an on-demand color. Please ensure liveness detection is completely disabled before disabling MPLS OAM."
- In Cisco NX-OS Release 10.1(2) SRTE OAM monitoring is supported for static policies and on-demand color having explicit path configured.
- The OAM sessions do not run for paths that are configured with dynamic option using PCEP.

Configuring MPLS OAM Monitoring

This section describes the CLIs required to enable proactive path monitoring for policies.

- **Global Configuration**

  This configuration enables OAM path monitoring for all configured policies.
- **Policy-specific Configuration**

  This configuration enables OAM path monitoring for a specific policy.

Global Configuration

Before you begin

You must ensure that the MPLS segment routing traffic engineering feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enters the segment routing configuration mode.

Step 3	**traffic-engineering**
	Example:
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		Enters the traffic engineering mode.

Step 4	[**liveness-detection**]
	Example:
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		Enters the liveness-detection configuration mode.

Step 5	**interval** *num*
	Example:
	`switch(config-sr-te-livedet)# interval 6000`
	`switch(config-sr-te-livedet)#`
		The duration of the interval in milliseconds. The default is 3000 ms.

Step 6	**multiplier** *num*
	Example:
	`switch(config-sr-te-livedet)# multiplier 5`
	`switch(config-sr-te-livedet)#`
		The multiplier sets the number of consecutive intervals that must fail for a path that is up to be considered down, and the number of consecutive intervals for a path that is down to be considered up. The default is 3.

Step 7	**mpls**
	Example:
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		Enables segment routing over mpls.

Step 8	[**no**]**oam**
	Example:
	`switch(config-sr-te-livedet-mpls)# oam`
	`switch(config-sr-te-livedet-mpls)#`
		Enable MPLS OAM Monitoring globally for all SRTE policies.
		The no form of this command disables OAM monitoring.

Step 9	**segment-list name** *sidlist-name* **Example:**	Creates the explicit SID list.

	Command or Action	Purpose

	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		Note This command has the autocomplete feature for the sidlist-name. To use this feature, add a question mark or press TAB.

Step 10	**policy** *policy name*
	Example:
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		Configures the policy.

Step 11	**color** *numberIP-end-point*
	Example:
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		Configures the color and the endpoint of the policy.

Step 12	**candidate-paths**
	Example:
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		Specifies the candidate paths for the policy.

Step 13	**preference** *preference-number*
	Example:
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 14	**explicit segment-list** *sidlist-name*
	Example:
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		Specifies the explicit list.
		Note This command has the autocomplete feature for the sidlist-name. To use this feature, add a question mark or press TAB.

Step 15	**on-demand color** *color_num*
	Example:
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		Enters the on-demand color template mode to configure an on-demand color for the specified color.

Step 16	**candidate-paths**
	Example:
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		Specifies the candidate paths for the policy.

	Command or Action	Purpose

Step 17	**preference** *preference-number*
	Example:
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 18	**explicit segment-list** *sidlist-name*
	Example:
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		Specifies the explicit list.
		Note This command has the autocomplete feature for the sidlist-name. To use this feature, add a question mark or press TAB.

Policy-specific Configuration

Before you begin

You must ensure that the MPLS segment routing traffic engineering feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enters the segment routing configuration mode.

Step 3	**traffic-engineering**
	Example:
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		Enters the traffic engineering mode.

Step 4	[**liveness-detection**]
	Example:
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		Enters the liveness-detection configuration mode.

Step 5	**interval** *num*
	Example:
	`switch(config-sr-te-livedet)# interval 6000`
	`switch(config-sr-te-livedet)#`
		The duration of the interval in milliseconds. The default is 3000 ms.

	Command or Action	Purpose

Step 6	**multiplier** *num*
	Example:
	`switch(config-sr-te-livedet)# multiplier 5`
	`switch(config-sr-te-livedet)#`
		The multiplier sets the number of consecutive intervals that must fail for a path that is up to be considered down, and the number of consecutive intervals for a path that is down to be considered up. The default is 3.

Step 7	**segment-list name** *sidlist-name*
	Example:
	`switch(config-sr-te)# segment-list name blue`
	`index 10 mpls label 16004`
	`index 10 mpls label 16005`
		Creates the explicit SID list.
		Note This command has the autocomplete feature for the sidlist-name. To use this feature, add a question mark or press TAB.

Step 8	**policy** *policy name*
	Example:
	`switch(config-sr-te)# policy 1`
	`switch(config-sr-te-pol)`
		Configures the policy.

Step 9	**color** *numberIP-end-point*
	Example:
	`switch(config-sr-te-pol)# color 1 endpoint 5.5.5.5`
	`switch(config-sr-te-pol)`
		Configures the color and the endpoint of the policy.

Step 10	**candidate-paths**
	Example:
	`switch(config-sr-te-pol)# candidate-paths`
	`switch(config-expcndpaths)#`
		Specifies the candidate paths for the policy.

Step 11	**preference** *preference-number*
	Example:
	`switch(config-expcndpaths)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 12	**explicit segment-list** *sidlist-name*
	Example:
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		Specifies the explicit list.
		Note This command has the autocomplete feature for the sidlist-name. To use this feature, add a question mark or press TAB.

Step 13	[**liveness-detection**]
	Example:
	`switch(config-sr-te)# liveness-detection`
	`switch(config-sr-te-livedet)#`
		Enters the liveness-detection configuration mode.

	Command or Action	Purpose

Step 14	[**no**]**index-limit** *num*
	Example:
	`switch(config-sr-te-livedet)# index-limit 20`
	`switch(config-sr-te-livedet)#`
		Monitors only SIDs that have an index of less than or equal to the user-specified number.

Step 15	[**no**]**shutdown**
	Example:
	`switch(config-sr-te-livedet)# shutdown`
	`switch(config-sr-te-livedet)#`
		Disables liveness detection. This is useful to temporarily disable liveness detection without entirely removing all related configuration.
		The no form of this command disables OAM monitoring.

Step 16	**mpls**
	Example:
	`switch(config-sr-te-livedet)# mpls`
	`switch(config-sr-te-livedet-mpls)#`
		Enables segment routing over mpls.

Step 17	[**no**]**oam**
	Example:
	`switch(config-sr-te-livedet-mpls)# oam`
	`switch(config-sr-te-livedet-mpls)#`
		Enable MPLS OAM Monitoring globally for all SRTE policies.
		The no form of this command disables OAM monitoring.

Step 18	**on-demand color** *color_num*
	Example:
	`switch(config-sr-te)# on-demand color 211`
	`switch(config-sr-te-color)#`
		Enters the on-demand color template mode to configure an on-demand color for the specified color.

Step 19	**candidate-paths**
	Example:
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		Specifies the candidate paths for the policy.

Step 20	**preference** *preference-number*
	Example:
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 21	**explicit segment-list** *sidlist-name*
	Example:
	`switch(cfg-pref)# explicit segment-list red`
	`switch(cfg-pref)#`
		Specifies the explicit list.
		Note This command has the autocomplete feature for the sidlist-name. To use this feature, add a question mark or press TAB.

	Command or Action	Purpose
Step 22	[**liveness-detection**] **Example:** `switch(config-sr-te-color)# liveness-detection switch(config-sr-te-color-livedet)#`	Enters the liveness-detection configuration mode.
Step 23	[**no**]**index-limit** *num* **Example:** `switch(config-sr-te-color-livedet)# index-limit 20 switch(config-sr-te-color-livedet)#`	Monitors only SIDs that have an index of less than or equal to the user-specified number.
Step 24	[**no**]**shutdown** **Example:** `switch(config-sr-te-color-livedet)# shutdown switch(config-sr-te-color-livedet)#`	Disables liveness detection. This is useful to temporarily disable liveness detection without entirely removing all related configuration. The no form of this command disables OAM monitoring.
Step 25	**mpls** **Example:** `switch(config-sr-te-color-livedet)# mpls switch(config-sr-te-color-livedet-mpls)#`	Enables segment routing over mpls.
Step 26	[**no**]**oam** **Example:** `switch(config-sr-te-color-livedet-mpls)# oam switch(config-sr-te-color-livedet-mpls)#`	Enable MPLS OAM Monitoring globally for all SRTE policies. The no form of this command disables OAM monitoring.

Verifying Configuration for MPLS OAM Monitoring

To display MPLS OAM monitoring configuration information, perform one of the following tasks:

Table 2: Verifying Configuration for MPLS OAM Monitoring

Command	Purpose
show srte policy	Displays only the authorized policies.
show srte policy [**all**]	Displays the list of all policies available in the SR-TE.
show srte policy [**detail**]	Displays the detailed view of all the requested policies.

Command	Purpose
show srte policy <*name*>	Filters the SR-TE policy with the name and displays the list of all policies available with that name in the SR-TE. **Note** This command has the autocomplete feature for the policy-name. To use this feature, add a question mark or press TAB.
show srte policy color <*color*> **endpoint** <*endpoint*>	Displays the SR-TE policy for the color and endpoint. **Note** This command has the autocomplete feature for color and endpoint. To use this feature, add a question mark or press TAB.
show srte policy proactive-policy-monitoring	Displays the list of all active proactive policy monitoring sessions that exist in the promon database. **Note** You can use the question mark option at the end of this command and provide one of the following options or press ENTER to display all the sessions: • brief - shows brief information about the sessions • color - shows the promon sessions related to the policy color • name - shows the promon sessions related to the policy name • session-id - shows the promon session for the session-id
show srte policy proactive-policy-monitoring [**brief**]	Displays only the list of session IDs and the states of the proactive policy monitoring sessions.
show srte policy proactive-policy-monitoring [**session** <*session-id*>]	Filters using session-id and displays information about that session in detail. **Note** This command has the autocomplete feature for the session-id. To use this feature, add a question mark or press TAB.

Command	Purpose
show srte policy proactive-policy-monitoring color <*color*> **endpoint**<*endpoint*>	Filters using color and endpoint and displays proactive policy monitoring sessions. **Note** This command has the autocomplete feature for color and endpoint. To use this feature, add a question mark or press TAB.

Configuration Example for MPLS OAM Monitoring

The following example shows how to configure MPLS OAM monitoring:

- Configuration example for global enablement with user specified multiplier and interval:

```
segment-routing
  traffic-engineering
    liveness-detection
        interval 6000
        multiplier 5
      mpls
        oam
    segment-list name blue
      index 10 mpls label 16004
      index 20 mpls label 16005
    segment-list name green
      index 10 mpls label 16003
      index 20 mpls label 16006
    segment-list name red
      index 10 mpls label 16002
      index 20 mpls label 16004
      index 30 mpls label 16005
    policy customer-1
      color 1 endpoint 5.5.5.5
      candidate-paths
        preference 100
          explicit segment-list red
    on-demand color 211
      candidate-paths
        preference 100
          explicit segment-list green
```

- Configuration example for policy enablement with user specified multiplier, interval, index-limit and shutdown option:

```
segment-routing
  traffic-engineering
    liveness-detection
        interval 6000
        multiplier 5
    segment-list name blue
      index 10 mpls label 16004
      index 20 mpls label 16005
    segment-list name green
      index 10 mpls label 16003
      index 20 mpls label 16006
    segment-list name red
      index 10 mpls label 16002
      index 20 mpls label 16004
```

```
  index 30 mpls label 16005
policy customer-1
  color 1 endpoint 5.5.5.5
  candidate-paths
    preference 100
      explicit segment-list red
  liveness-detection
    index-limit 20
    shutdown
    mpls
      oam
on-demand color 211
  candidate-paths
    preference 100
      explicit segment-list green
  liveness-detection
      index-limit 20
      shutdown
      mpls
        oam
```

Configuring Egress Peer Engineering with Segment Routing

BGP Prefix SID

In order to support segment routing, BGP requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP prefix SID is always global within the segment routing BGP domain and identifies an instruction to forward the packet over the ECMP-aware best path computed by BGP to the related prefix. The BGP prefix SID identifies the BGP prefix segment.

Adjacency SID

The adjacency segment Identifier (SID) is a local label that points to a specific interface and a next hop out of that interface. No specific configuration is required to enable adjacency SIDs. Once segment routing is enabled over BGP for an address family, for any interface that BGP runs over, the address family automatically allocates an adjacency SID toward every neighbor out of that interface.

High Availability for Segment Routing

In-service software upgrades (ISSUs) are minimally supported with BGP graceful restart. All states (including the segment routing state) must be relearned from the BGP router's peers. During the graceful restart period, the previously learned route and label state are retained.

Overview of BGP Egress Peer Engineering With Segment Routing

Cisco Nexus 9000 Series switches are often deployed in massive scale data centers (MSDCs). In such environments, there is a requirement to support BGP Egress Peer Engineering (EPE) with Segment Routing (SR).

Segment Routing (SR) leverages source routing. A node steers a packet through a controlled set of instructions, known as segments, by prepending the packet with an SR header. A segment can represent any topological

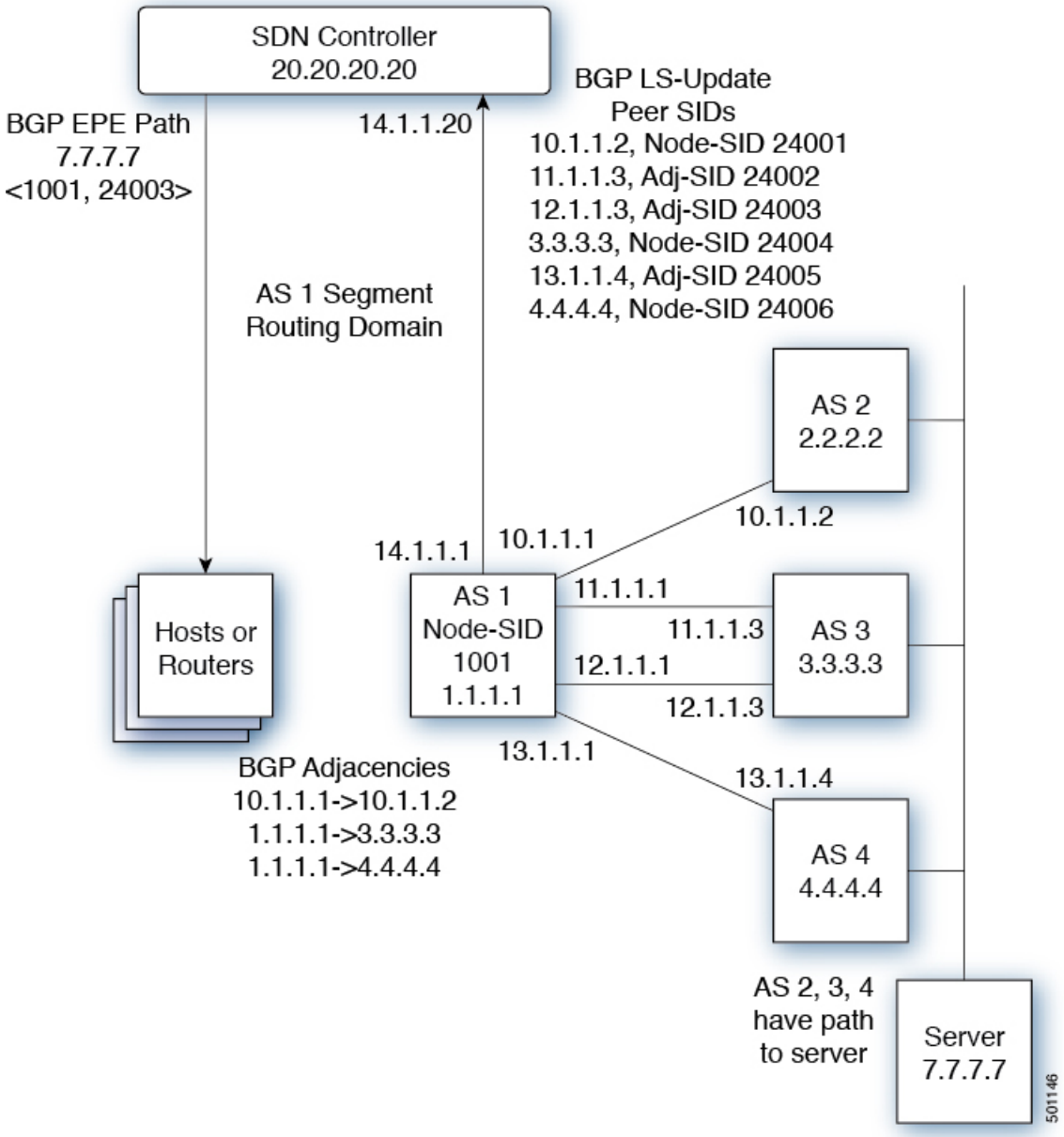
or service-based instruction. SR allows steering a flow through any topological path or any service chain while maintaining per-flow state only at the ingress node of the SR domain. For this feature, the Segment Routing architecture is applied directly to the MPLS data plane.

In order to support Segment Routing, BGP requires the ability to advertise a Segment Identifier (SID) for a BGP prefix. A BGP prefix is always global within the SR or BGP domain and it identifies an instruction to forward the packet over the ECMP-aware best-path that is computed by BGP to the related prefix. The BGP prefix is the identifier of the BGP prefix segment.

The SR-based Egress Peer Engineering (EPE) solution allows a centralized (SDN) controller to program any egress peer policy at ingress border routers or at hosts within the domain.

In the following example, all three routers run iBGP and they advertise NRLI to one another. The routers also advertise their loopback as the next-hop and it is recursively resolved. This provides an ECMP between the routers as displayed in the illustration.

Figure 3: Example of Egress Peer Engineering



The SDN controller receives the Segment IDs from the egress router 1.1.1.1 for each of its peers and adjacencies. It can then intelligently advertise the exit points to the other routers and the hosts within the controller's routing domain. As displayed in the illustration, the BGP Network Layer Reachability Information (NLRI) contains both the Node-SID to Router 1.1.1.1 and the Peer-Adjacency-SID 24003 indicating that the traffic to 7.7.7.7 should egress over the link 12.1.1.1->12.1.1.3.

Guidelines and Limitations for BGP Egress Peer Engineering

BGP Egress Peer Engineering has the following guidelines and limitations:

- BGP Egress Peer Engineering is only supported for IPv4 BGP peers. IPv6 BGP peers are not supported.
- BGP Egress Peer Engineering is only supported in the default VPN Routing and Forwarding (VRF) instance.
- Any number of Egress Peer Engineering (EPE) peers may be added to an EPE peer set. However, the installed resilient per-CE FEC is limited to 32 peers.
- A given BGP neighbor can only be a member of a single peer-set. Peer-sets are configured. Multiple peer-sets are not supported. An optional **peer-set** name may be specified to add neighbor to a peer-set. The corresponding RPC FEC load-balances the traffic across all the peers in the peer-set. The peer-set name is a string that is a maximum length of 63 characters (64 NULL terminated). This length is consistent with the NX-OS policy name lengths. A peer can only be a member of a single peer-set.
- Adjacencies for a given peer are not separately assignable to different peer-sets.
- Beginning with Cisco NX-OS Release 9.3(3), BGP Egress Peer Engineering is supported on Cisco Nexus 9300-GX platform switches.

Configuring Neighbor Egress Peer Engineering Using BGP

With the introduction of RFC 7752 and draft-ietf-idr-bgpls-segment-routing-epe, you can configure Egress Engineering. The feature is valid only for external BGP neighbors and it is not configured by default. Egress Engineering uses RFC 7752 encoding.

Before you begin

- You must enable BGP.
- After an upgrade from Release 7.0(3)I3(1) or Release 7.0(3)I4(1), configure the TCAM region before configuring Egress Peer Engineering (EPE) on Cisco Nexus 9000 Series switches using the following commands:
  1. switch# **hardware access-list tcam region vpc-convergence 0**
  2. switch# **hardware access-list tcam region racl 0**
  3. switch# **hardware access-list tcam region mpls 256 double-wide**
- Save the configuration and reload the switch.

For more information, see the Using Templates to Configure ACL TCAM Region Sizes and Configuring ACL TCAM Region Sizes sections in the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**router bgp** <*bgp autonomous number*>	Specifies the autonomous router BGP number.
Step 3	**neighbor** <*IP address*>	Configures the IP address for the neighbor.

Step 4	[**no**|**default**] **egress-engineering** [**peer-set** *peer-set-name*]
	Example:
	`switch(config)# router bgp 1`
	`switch(config-router)# neighbor 4.4.4.4`
	`switch(config-router)# egress-engineering`
	`peer-set NewPeer`
		Specifies whether a Peer-Node-SID is allocated for the neighbor and it is advertised in an instance of a BGP Link-State (BGP-LS) address family Link NLRI. If the neighbor is a multi-hop neighbor, a BGP-LS Link NLRI instance is also advertised for each Equal-Cost-MultiPath (ECMP) path to the neighbor and it includes a unique Peer-Adj-SID.
		Optionally, you can add the neighbor to a peer-set. The Peer-Set-SID is also advertised in the BGP-LS Link NLRI in the same instance as the Peer-Node-SID. BGP Link-State NLRI is advertised to all neighbors with the link-state address family configured.
		See RFC 7752 and draft-ietf-idr-bgpls-segment-routing-epe-05 for more information on EPE.

Configuration Example for Egress Peer Engineering

See the Egress Peer Engineering sample configuration for the BGP speaker 1.1.1.1. Note that the neighbor 20.20.20.20 is the SDN controller.

```
 hostname epe-as-1
install feature-set mpls
feature-set mpls

feature telnet
feature bash-shell
feature scp-server
feature bgp
feature mpls segment-routing

segment-routing mpls
vlan 1

vrf context management
  ip route 0.0.0.0/0 10.30.97.1
  ip route 0.0.0.0/0 10.30.108.1

interface Ethernet1/1
  no switchport
  ip address 10.1.1.1/24
  no shutdown

interface Ethernet1/2
  no switchport
```

```
  ip address 11.1.1.1/24
  no shutdown

interface Ethernet1/3
  no switchport
  ip address 12.1.1.1/24
  no shutdown

interface Ethernet1/4
  no switchport
  ip address 13.1.1.1/24
  no shutdown

interface Ethernet1/5
  no switchport
  ip address 14.1.1.1/24
  no shutdown

interface mgmt0
  ip address dhcp
  vrf member management


interface loopback1
  ip address 1.1.1.1/32
line console

line vty
ip route 2.2.2.2/32 10.1.1.2
ip route 3.3.3.3/32 11.1.1.3
ip route 3.3.3.3/32  12.1.1.3
ip route 4.4.4.4/32  13.1.1.4
ip route 20.20.20.20/32 14.1.1.20

router bgp 1
  address-family ipv4 unicast
  address-family link-state
 neighbor 10.1.1.2
    remote-as 2
    address-family ipv4
    egress-engineering
 neighbor 3.3.3.3
   remote-as 3
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
 neighbor 4.4.4.4
   remote-as 4
   address-family ipv4
   update-source loopback1
   ebgp-multihop 2
   egress-engineering
neighbor 20.20.20.20
   remote-as 1
   address-family link-state
   update-source loopback1
   ebgp-multihop 2
neighbor 124.11.50.5
    bfs
    remote-as 6
    update-source port-channel50.11
    egress-engineering peer-set pset2 <<<<<<<
    address-family ipv4 unicast
```

```
neighbor 124.11.101.2
    bfd
    remote-as 6
    update-source Vlan2401
    egress-engineering
    address-family ipv4 unicast
```

This example shows sample output for the **show bgp internal epe** command.

```
switch# show bgp internal epe
BGP Egress Peer Engineering (EPE) Information:
Link-State Server: Inactive
Link-State Client: Active
Configured EPE Peers: 26
Active EPE Peers: 3
EPE SID State:
RPC SID Peer or Set Assigned
ID Type Set Name ID Label Adj-Info, iod
1 Node 124.1.50.5 1 1600
2 Set pset1 2 1601
3 Node 6.6.6.6 3 1602
4 Node 124.11.50.5 4 1603
5 Set pset2 5 1604
6 Adj 6.6.6.6 6 1605 124.11.50.4->124.11.50.5/0x1600b031, 80
7 Adj 6.6.6.6 7 1606 124.1.50.4->124.1.50.5/0x16000031, 78
EPE Peer-Sets:
IPv4 Peer-Set: pset1, RPC-Set 2, Count 7, SID 1601
Peers: 124.11.116.2 124.11.111.2 124.11.106.2 124.11.101.2
124.11.49.5 124.1.50.5 124.1.49.5
IPv4 Peer-Set: pset2, RPC-Set 5, Count 5, SID 1604
Peers: 124.11.117.2 124.11.112.2 124.11.107.2 124.11.102.2
124.11.50.5
IPv4 Peer-Set: pset3, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.118.2 124.11.113.2 124.11.108.2 124.11.103.2
IPv4 Peer-Set: pset4, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.119.2 124.11.114.2 124.11.109.2 124.11.104.2
IPv4 Peer-Set: pset5, RPC-Set 0, Count 4, SID unspecified
Peers: 124.11.120.2 124.11.115.2 124.11.110.2 124.11.105.2
switch#
```

Configuring the BGP Link State Address Family

You can configure the BGP link state address family for a neighbor session with a controller to advertise the corresponding SIDs. You can configure this feature in global configuration mode and neighbor address family configuration mode.

Before you begin

You must enable BGP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**router bgp** <*bgp autonomous number*>	Specifies the autonomous router BGP number.

Step 3	[**no**] **address-family link-state**
	Example:
	`switch(config)# router bgp 64497`
	`switch (config-router af)# address-family link-state`
		Enters address-family interface configuration mode.
		Note This command can also be configured in neighbor address-family configuration mode.

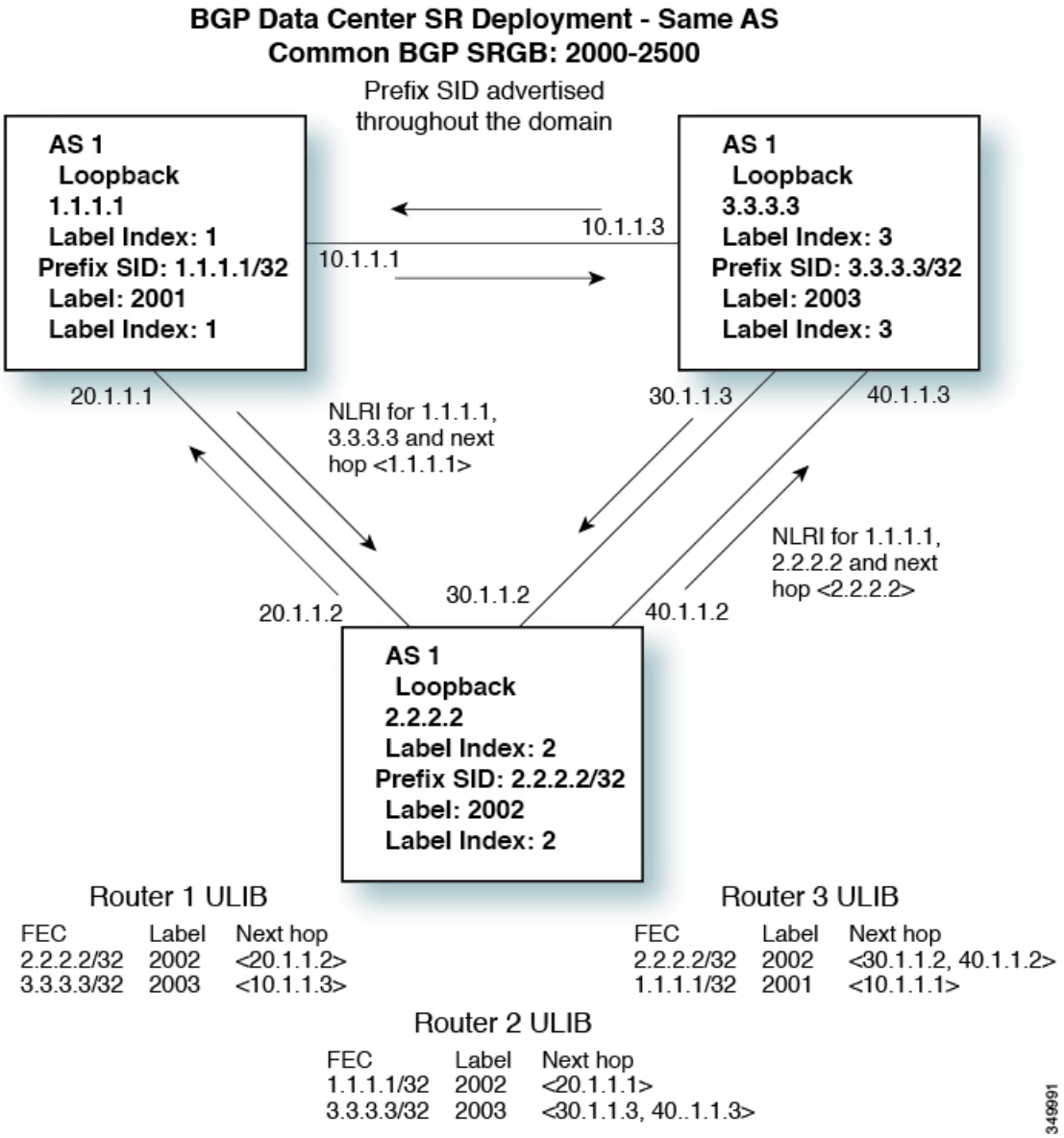
Step 4	**neighbor** <*IP address*>	Configures the IP address for the neighbor.

Step 5	[**no**] **address-family link-state**
	Example:
	`switch(config)#router bgp 1`
	`switch(config-router)#address-family link-state`
	`switch(config-router)#neighbor 20.20.20.20`
	`switch(config-router)#address-family link-state`
		Enters address-family interface configuration mode.
		Note This command can also be configured in neighbor address-family configuration mode.

BGP Prefix SID Deployment Example

In the simple example below, all three routers are running iBGP and advertising Network Layer Reachability Information (NRLI) to one another. The routers are also advertising their loopback interface as the next hop, which provides the ECMP between routers 2.2.2.2 and 3.3.3.3.

Figure 4: BGP Prefix SID Simple Example



Configuring Layer2 EVPN over Segment Routing MPLS

About Layer 2 EVPN

Ethernet VPN (EVPN) is a next generation solution that provides ethernet multipoint services over MPLS networks. EVPN operates in contrast to the existing Virtual Private LAN Service (VPLS) by enabling control-plane based MAC learning in the core. In EVPN, PEs participating in the EVPN instances learn customer MAC routes in control-plane using MP-BGP protocol. Control-plane MAC learning brings several

benefits that allow EVPN to address the VPLS shortcomings, including support for multihoming with per-flow load balancing.

In a data center network, the EVPN control plane provides:

- Flexible workload placement that is not restricted with the physical topology of the data center network. Therefore, you can place virtual machines (VM) anywhere within the data center fabric.
- Optimal East-West traffic between servers within and across data centers. East-West traffic between servers, or virtual machines, is achieved by most specific routing at the first hop router. First hop routing is done at the access layer. Host routes must be exchanged to ensure most specific routing to and from servers or hosts. VM mobility is supported by detecting new endpoint attachment when a new MAC address or the IP address is directly connected to the local switch. When the local switch sees the new MAC or the IP address, it signals the new location to rest of the network.
- Segmentation of Layer 2 and Layer 3 traffic, where traffic segmentation is achieved using MPLS encapsulation and the labels (per-BD label and per-VRF labels) act as the segment identifier.

Guidelines and Limitations for Layer 2 EVPN over Segment Routing MPLS

Layer 2 EVPN over segment routing MPLS has the following guidelines and limitations:

- Segment routing Layer 2 EVPN flooding is based on the ingress replication mechanism. MPLS core does not support multicast.
- ARP suppression is not supported.
- Consistency checking on vPC is not supported.
- The same Layer 2 EVI and Layer 3 EVI cannot be configured together.
- Beginning with Cisco NX-OS Release 9.3(1), Layer 2 EVPN is supported on Cisco Nexus 9300-FX2 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), Layer 2 EVPN over segment routing MPLS is supported on Cisco Nexus 9300-GX and Cisco Nexus 9300-FX3 platform switches.

Configuring Layer 2 EVPN over Segment Routing MPLS

Before you begin

Do the following:

- You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.
- You must enable the MPLS segment routing feature.
- You must enable the nv overlay feature using the **nv overlay** command.
- You must enable EVPN control plane using the **nv overlay evpn** command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp** **Example:** `switch(config)#feature bgp`	Enables BGP feature and configurations.
Step 3	**install feature-set mpls** **Example:** `switch(config)#install feature-set mpls`	Enables MPLS configuration commands.
Step 4	**feature-set mpls** **Example:** `switch(config)#install feature-set mpls`	Enables MPLS configuration commands.
Step 5	**feature mpls segment-routing** **Example:** `switch(config)#feature mpls segment-routing`	Enables segment routing configuration commands.
Step 6	**feature mpls evpn** **Example:** `switch(config)#feature mpls evpn`	Enables EVPN over MPLS configuration commands. This command is mutually exclusive with the **feature-nv** CLI command.
Step 7	**feature nv overlay** **Example:** `switch(config)#feature nv overlay`	Enables the NVE feature that is used for the segment routing Layer 2 EVPN.
Step 8	**nv overlay evpn** **Example:** `switch(config)#nv overlay evpn`	Enables EVPN.
Step 9	**interface loopback** *Interface_Number* **Example:** `switch(config)#interface loopback 1`	Configures the loopback interface for NVE.
Step 10	**ip address** *address* **Example:** `switch(config-if)#ip address 192.168.15.1`	Configures the IP address.

	Command or Action	Purpose
Step 11	**exit** **Example:** `switch(config-if)#exit`	Exits global address family configuration mode.
Step 12	**evpn** **Example:** `switch(config)#evpn`	Enters the EVPN configuration mode.

Step 13	**evi** *number*
	Example:
	`switch(config-evpn)#evi 1000`
	`switch(config-evpn-sr)#`
		Configures Layer 2 EVI. If required, you can manually configure the RT based on the EVI that is generated automatically.

Step 14	**encapsulation mpls** **Example:** `switch(config-evpn)#encapsulation mpls`	Enables MPLS encapsulation and ingress-replication.
Step 15	**source-interface loopback** *Interface_Number* **Example:** `switch(config-evpn-nve-encap)#source-interface loopback 1`	Specifies the NVE source interface.
Step 16	**exit** **Example:** `switch(config-evpn-nve-encap)#exit`	Exits the configuration.
Step 17	**vrf context** *VRF_NAME* **Example:** `switch(config)#vrf context Tenant-A`	Configures the VRF.
Step 18	**evi** *EVI_ID* **Example:** `switch(config-vrf)#evi 30001`	Configures L3 EVI.
Step 19	**exit** **Example:** `switch(config-vrf)#exit`	Exits the configuration.
Step 20	**VLAN** *VLAN_ID* **Example:** `switch(config)#vlan 1001`	Configures VLAN.
Step 21	**evi auto** **Example:**	Configures L2 EVI.

	Command or Action	Purpose
	`switch(config-vlan)#evi auto`	
Step 22	**exit** **Example:** `switch(config-vlan)#exit`	
Step 23	**router bgp** *autonomous-system-number* **Example:** `switch(config)#router bgp 1`	Enters the BGP configuration mode.
Step 24	**address-family l2vpn evpn** **Example:** `switch(config-router)#address-family l2vpn evpn`	Enables EVPN address family globally.
Step 25	**neighbor address** *remote-as autonomous-system-number* **Example:** `switch(config-router)#neighbor 192.169.13.1 remote as 2`	Configures BGP neighbor.
Step 26	**address-family l2vpn evpn** **Example:** `switch(config-router-neighbor)#address-family l2vpn evpn`	Enables EVPN address family for neighbor.
Step 27	**encapsulation mpls** **Example:** `switch(config-router-neighbor)#encapsulation mpls`	Enables MPLS encapsulation.
Step 28	**send-community extended** **Example:** `switch(config-router-neighbor)#send-community extended`	Configures BGP to advertise extended community lists.
Step 29	**vrf** *VRF_NAME* **Example:** `switch(config-router)#vrf Tenant-A`	Configures BGP VRF.
Step 30	**exit** **Example:** `switch(config-router)#exit`	Exits the configuration.

Configuring VLAN for EVI

Procedure

	Command or Action	Purpose
Step 1	**vlan** *number*	Specifies the VLAN.
Step 2	**evi** *auto*	Creates a BD label for the VLAN. This label is used as an identifier for the VLAN across the segment routing Layer 2 EVPN.

Configuring the NVE Interface

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 1`	Associates the IP address with this loopback interface and uses this IP address for the segment routing configuration.
Step 3	**ip address** **Example:** `switch(config-if)#ip address 192.169.15.1/32`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 4	**evpn** **Example:** `switch(config)#evpn`	Enters EVPN configuration mode.
Step 5	**encapsulation mpls** **Example:** `switch(config-evpn)# encapsulation mpls`	Enables MPLS encapsulation and ingress-replication.
Step 6	**source-interface** *loopback_number* **Example:** `switch(config-evpn-nve-encap)#source-interface loopback 1`	Specifies the NVE source interface.
Step 7	**exit** **Example:**	Exits segment routing mode and returns to the configuration terminal mode.

	Command or Action	Purpose
	`switch(config)# exit`	

Configuring EVI Under VRF

Procedure

	Command or Action	Purpose
Step 1	**vrf context** *tenant*	Create a VRF Tenant.
Step 2	**evi** *number*	Configure Layer 3 EVI under VRF.

Configuring Anycast Gateway

The fabric forwarding configuration is necessary only if the SVIs are configured in the anycast mode.

Procedure

	Command or Action	Purpose
Step 1	**fabric forwarding anycast-gateway-mac 0000.aabb.ccdd**	Configures the distributed gateway virtual MAC address.
Step 2	**fabric forwarding mode anycast-gateway**	Associates SVI with the Anycast Gateway under the interface configuration mode.

Advertising Labelled Path for the Loopback Interface

The loopback interface, advertised as Layer 2 EVPN endpoint should be mapped to a label index. Thereby BGP advertises MPLS labelled path for the same.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**]**router ospf** *process* **Example:** `switch(config)# router ospf test`	Enables the OSPF mode.
Step 3	**segment-routing** **Example:**	Configures the segment routing functionality under OSPF.

	Command or Action	Purpose
	`switch(config-router)# segment-routing mpls`	
Step 4	**connected-prefix-sid-map** **Example:** `switch(config-sr-mpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.
Step 5	**address-family ipv4** **Example:** `switch(config-sr-mpls-conn)# address-family ipv4`	Specifies IPv4 address prefixes.
Step 6	**1.1.1.1/32 index 100** **Example:** `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 100`	Associates SID 100 with the address 1.1.1.1/32.
Step 7	**exit-address-family** **Example:** `switch(config-sr-mpls-conn-af)# exit-address-family`	Exits the address family.

About SRv6 Static Per-Prefix TE

The SRv6 Static Per-Prefix TE feature allows you to map and advertise prefixes that at mapped to non-default VRFs. This feature allows you to advertise multiple prefixes in a single instance using the matching VRF route target and prevents the manual entry of each prefix.

In Cisco NX-OS Release 9.3(5), only one VNF can service a VM.

Configuring a SRv6 Static Per-Prefix TE

Before you begin

Do the following:

- You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.
- You must enable the MPLS segment routing feature.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**vrf context** *VRF_Name* **Example:** `switch(config)# vrf context vrf_2_7_8`	Defines VRF and enters the VRF configuration mode.
Step 3	**rd** *rd_format* **Example:** `switch(config-vrf)# rd 2.2.2.0:2`	Assign the RD to VRF.
Step 4	**address-family** {**ipv4** | **ipv6** } **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Specifies either the IPv4 or the IPv6 address family for the VRF instance and enters the address family configuration mode.
Step 5	**route-target import** *route-target-id* **Example:** `switch(config-vrf)# route-target import 1:2`	Configures the importing of routes to the VRF.
Step 6	**route-target import** *route-target-id* **evpn** **Example:** `switch(config-vrf)# route-target import 1:2 evpn`	Configures importing of routes that have a matching route target value from the Layer 3 EVPN to the VRF.
Step 7	**route-target export** *route-target-id* **Example:** `switch(config-vrf)# route-target export 1:2`	Configures the exporting of routes from the VRF.
Step 8	**route-target export** *route-target-id* **evpn** **Example:** `switch(config-vrf)# route-target export 1:2 evpn`	Configures exporting of routes that have a matching route target value from the VRF to the Layer 3 EVPN.
Step 9	**router bgp** *autonomous-system-number* **Example:** `switch(config)# router bgp 65000`	Enables BGP and assigns the AS number to the local BGP speaker.
Step 10	**router-id** *id* **Example:** `switch(config-router)# router-id 2.2.2.0`	Configures the router ID.
Step 11	**address-family l2vpn evpn** **Example:**	Enters global address family configuration mode for the Layer 2 VPN EVPN.

	Command or Action	Purpose
	`switch(config-router-af)# address-family l2vpn evpn`	

Step 12	**neighbor** *ipv4-address* **remote-as**
	Example:
	`switch(config-router)# neighbor 7.7.7.0 remote-as 65000`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP peer.

Step 13	**update-source loopback** *number* **Example:** `switch(config-router-neighbor)# update-source loopback0`	Specifies the loopback number.
Step 14	**address-family l2vpn evpn** **Example:** `switch(config-router-neighbor)#address-family l2vpn evpn`	Enables EVPN address family for a neighbor.
Step 15	**send-community extended** **Example:** `switch(config-router-neighbor)#send-community extended`	Configures BGP to advertise extended community lists.
Step 16	**encapsulation mpls** **Example:** `switch(config-router-neighbor)#encapsulation mpls`	Enables MPLS encapsulation.
Step 17	**exit** **Example:** `switch(config-router-neighbor)#exit`	Exits the configuration.

Example

The following example shows how to configure RPM configuration in order to define the VRF VT.

```
rf context vrf_2_7_8
          rd 2.2.2.0:2
          address-family ipv4 unicast
            route-target import 0.0.1.1:2
            route-target import 0.0.1.1:2 evpn
            route-target export 0.0.1.1:2
            route-target export 0.0.1.1:2 evpn
ip extcommunity-list standard vrf_2_7_8-test permit rt 0.0.1.1:2
            route-map Node-2 permit 4
            match extcommunity vrf_2_7_8-test
            set extcommunity color 204
```

About RD Auto

The auto-derived Route Distinguisher (rd auto) is based on the Type 1 encoding format as described in IETF RFC 4364 section 4.2 https://tools.ietf.org/html/rfc4364#section-4.2. The Type 1 encoding allows a 4-byte administrative field and a 2-byte numbering field. Within Cisco NX-OS, the auto derived RD is constructed with the IP address of the BGP Router ID as the 4-byte administrative field (RID) and the internal VRF identifier for the 2-byte numbering field (VRF ID).

The 2-byte numbering field is always derived from the VRF, but results in a different numbering scheme depending on its use for the IP-VRF or the MAC-VRF:

- The 2-byte numbering field for the IP-VRF uses the internal VRF ID starting at 1 and increments. VRF IDs 1 and 2 are reserved for the default VRF and the management VRF respectively. The first custom defined IP VRF uses VRF ID 3.
- The 2-byte numbering field for the MAC-VRF uses the VLAN ID + 32767, which results in 32768 for VLAN ID 1 and incrementing.

Example auto-derived Route Distinguisher (RD)

- IP-VRF with BGP Router ID 192.0.2.1 and VRF ID 6 - RD 192.0.2.1:6
- MAC-VRF with BGP Router ID 192.0.2.1 and VLAN 20 - RD 192.0.2.1:32787

About Route-Target Auto

The auto-derived Route-Target (route-target import/export/both auto) is based on the Type 0 encoding format as described in IETF RFC 4364 section 4.2 (https://tools.ietf.org/html/rfc4364#section-4.2). IETF RFC 4364 section 4.2 describes the Route Distinguisher format and IETF RFC 4364 section 4.3.1 refers that it is desirable to use a similar format for the Route-Targets. The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative field and the Service Identifier (EVI) for the 4-byte numbering field.

2-byte ASN

The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto-derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative filed and the Service Identifier (EVI) for the 4-byte numbering field.

Examples of an auto derived Route-Target (RT):

- IP-VRF within ASN 65001 and L3EVI 50001 - Route-Target 65001:50001
- MAC-VRF within ASN 65001 and L2EVI 30001 - Route-Target 65001:30001

For Multi-AS environments, the Route-Targets must either be statically defined or rewritten to match the ASN portion of the Route-Targets.

Note Auto derived Route-Targets for a 4-byte ASN are not supported.

4-byte ASN

The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto-derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative filed and the Service Identifier (EVI) for the 4-byte numbering field. With the ASN demand of 4-byte length and the EVI requiring 24-bit (3-bytes), the Sub-Field length within the Extended Community is exhausted (2-byte Type and 6-byte Sub-Field). As a result of the length and format constraint and the importance of the Service Identifiers (EVI) uniqueness, the 4-byte ASN is represented in a 2-byte ASN named AS_TRANS, as described in IETF RFC 6793 section 9 (https://tools.ietf.org/html/rfc6793#section-9). The 2-byte ASN 23456 is registered by the IANA (https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml) as AS_TRANS, a special purpose AS number that aliases 4-byte ASNs.

Example auto derived Route-Target (RT) with 4-byte ASN (AS_TRANS):

- IP-VRF within ASN 65656 and L3EVI 50001 - Route-Target 23456:50001
- MAC-VRF within ASN 65656 and L2EVI 30001 - Route-Target 23456:30001

Configuring RD and Route Targets for BD

The Bridge Domain (BD) RD and Route Targets are automatically generated when you configure **evi auto** under the VLAN. To configure the BD RD and Route Targets manually, perform these steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**evpn** **Example:** `switch(config)# evpn`	Enters EVPN configuration mode.
Step 3	**evi** *VLAN_ID* **Example:** `switch(config-evpn)# evi 1001`	Specifies L2 EVI to configure RD/Route Target.
Step 4	**rd** *rd_format* **Example:** `switch(config-evpn-evi-sr)# rd 192.1.1.1:33768`	Configures RD.
Step 5	**route-target both** *rt_format* **Example:** `switch(config-evpn-evi-sr)# route-target both 1:20001`	Configures Route Target.

Configuring RD and Route Targets for VRF

The VRF RD and Route Targets are automatically generated when you configure the **evi** *evi_ID* under the VRF. To configure the VRF RD and Route Targets manually, perform these steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vrf context** *VRF_NAME* **Example:** `switch(config)# vrf context A`	Configures the VRF.
Step 3	**rd auto** or **rd_format** **Example:** `switch(config-vrf)# rd auto`	Configures RD.
Step 4	**address-family ipv4 unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Enables IPv4 address family.
Step 5	**route-target both** *rt_format* **evpn** **Example:** `switch(config-vrf-af-ipv4)# route-target both 1:30001 evpn`	Configures Route Target.

Configuration Examples for Layer 2 EVPN over Segment Routing MPLS

The following examples show the configuration for Layer 2 EVPN over Segment Routing MPLS:

```
install feature-set mpls
feature-set mpls
nv overlay evpn
feature bgp
feature mpls segment-routing
feature mpls evpn
feature interface-vlan
feature nv overlay

fabric forwarding anycast-gateway-mac 0000.1111.2222

vlan 1001
  evi auto

vrf context Tenant-A
  evi 30001
```

```
interface loopback 1
  ip address 192.168.15.1/32

interface vlan 1001
  no shutdown
  vrf member Tenant-A
  ip address 111.1.0.1/16
  fabric forwarding mode anycast-gateway

router bgp 1
  address-family l2vpn evpn
    neighbor 192.169.13.1
      remote-as 2
      address-family l2vpn evpn
        send-community extended
        encapsulation mpls
    vrf Tenant-A

evpn
  encapsulation mpls
    source-interface loopback 1
```

Configuring Proportional Multipath for VNF for Segment Routing

About Proportional Multipath for VNF for Segment Routing

In Network Function Virtualization Infrastructures (NFVi), service networks (Portable IPs) are routed by Virtual Network Functions (VNFs). The VNFs, also referred to as portable IP-Gateway (PIP-GW) routes the data packets to and from the VMs in the VNF. The Proportional Multipath for VNF for Segment Routing feature enables advertising the VNF of a service network (PIP) in the EVPN address-family. The IP address of the VNF is encoded in the “Gateway-IP Address” field of the EVPN IP Prefix Route NLRI advertisement of a service network.

By advertising the IP address of the VNFs, ingress nodes in the EVPN fabric recursively resolve the VNF IP address to the leaf attached to the VNF, which could be the same node that advertises the service network (PIP).

Route-injectors are BGP protocols that inject routes in the IPv4 or IPv6 AF. In this case, the route-injector injects routes to the VMs whose next hop is set as VNFs.

Unlike a route-injector, VNFs can participate in a routing protocol to advertise the VM reachability. The supported protocols are eBGP, IS-IS, and OSPF.

Enabling Proportional Multipath for VNF for Segment Routing

You can enable the Proportional Multipath for VNF for Segment Routing feature to redistribute routes for IGP or static routes by preserving the next-hop paths. You can then export and advertise the gateway-IP for the reoriginated EVPN type-5 routes.

In Cisco NX-OS Release 9.3(5), only one VNF can service a VM.

Before you begin

Do the following:

- Install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.
- Enable the MPLS segment routing feature.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enter global configuration mode.

Step 2	**route-map** *export-l2evpn-rtmap* **permit** *10* **Example:** `switch(config)# route-map export-l2evpn-rtmap permit 10`	<<need description>>
Step 3	**match ip address prefix-list** *pip-pfx-list* **Example:** `switch(config-route-map)# match ip prefix-list vm-pfx-list`	Defines the prefixes that must be advertised with PIP-GW as the gateway.
Step 4	**set evpn gateway-ip use-nexthop** **Example:** `switch(config-route-map)# set evpn gateway-ip use-nexthop`	Defines specific routes to advertise the gateway-ip.

Step 5	**vrf context** *VRF_Name*
	Example:
	`switch(config-route-map)# vrf context vrf`
	`switch(config-route-map)# address-family ipv4 unicast`
	`switch(config-route-map)# export map export-l2evpn-rtmap`
		Applies the route map to the vrf context.

Step 6	**address-family ipv4 unicast**
	Example:
	`switch(config-route-map)# address-family ipv4 unicast`
	`switch(config-route-map)# export map export-l2evpn-rtmap`
		Applies the route map to the vrf context.

Step 7	**export map** *export-l2evpn-rtmap* **Example:** `switch(config-route-map)# export map export-l2evpn-rtmap`	Applies the route map to the vrf context.

	Command or Action	Purpose
Step 8	**router bgp** *number* **Example:** `switch(config)# router bgp 100`	Configure BGP.
Step 9	**vrf** *VRF_Name* **Example:** `switch(config-route-map)# vrf vrf3`	Applies the route map to the vrf context.
Step 10	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configure address family for IPv4.
Step 11	**export-gateway-ip** **Example:** `switch(config-route-map)# export-gateway-ip`	Exports and advertises the gateway-ip to reconnect the EVPN type-5 routes. **Note** The export gateway-ip and set the EVPN gateway configurations can be performed simultaneously. If you configure them simultaneously, all prefixes are exported with the gateway-ip.

vPC Multihoming

About Multihoming

Cisco Nexus platform switches support vPC-based multihoming, where a pair of switches act as a single device for redundancy and both switches function in active mode. With Cisco Nexus platform switches in an EVPN environment, there are two solutions that support Layer 2 multihoming; these solutions are based on the traditional vPC (emulated or virtual IP address), where the MCT link is required and the BGP EVPN techniques.

While using the BGP EVPN control plane, each vPC pair uses a common virtual IP (VIP) to provide active/active redundancy. BGP EVPN based multihoming further provides fast convergence during certain failure scenarios, that otherwise cannot be achieved without a control protocol (data plane flood and learn).

Per-BD label on vPC Peers

To ensure that the vPC peers have the same per-BD label, you must specify the per-BD label to have the following value:

```
Label value = Label_base + VLAN_ID
```

The label base is configured on the same vPC peers. Currently, the VLAN configuration is identical on both the vPC peers, which ensures that both vPC peers have the same label.

In Cisco NX-OS Release 9.3(1), configuring the per-BD label is not supported. This release supports only evi auto.

Per-VRF label on vPC Peers

To ensure that the vPC peers have the same per-VRF label, you must specify the per-VRF label to have the following value:

```
Label value = Label_base + vrf_allocate_index
```

To configure the allocate-index for the vPC peers, do the following:

```
Router bgp 1
  vrf Tenant_A
    allocate-index 11
```

Configuring Backup Link

The backup link needs to be configured between the vPC peers. This link can be any Layer 3 link which is parallel to MCT.

Example

```
interface vlan 100
    ip add 10.1.1.1/24
        mpls ip forwarding

< enable underlay protocol >
```

Guidelines and Limitations for vPC Multihoming

vPC multihoming has the following guidelines and limitations:

- ESI-based multihoming is not supported.
- The physical and virtual secondary IP addresses should be both advertised via the MPLS labeled path.
- vPC consistency checking is not supported for the per-BD label configuration.

Configuration Examples for vPC Multihoming

This example shows the configuration for vPC multihoming:

- vPC Primary

```
interface loopback1
  ip address 192.169.15.1/32
  ip address 192.169.15.15/32 secondary

evpn
  encapsulation mpls
    source-interface loopback1

vlan 101
  evi auto

vrf context A
  evi 301
```

```
router bgp 1
  vrf A
    allocate-index 1001
```

- vPC Secondary

```
interface loopback1
  ip address 192.169.15.2/32
  ip address 192.169.15.15/32 secondary

evpn
  encapsulation mpls
    source-interface loopback1

vlan 101
  evi auto

vrf context A
  evi 301

router bgp 1
  vrf A
    allocate-index 1001
```

Configuring Layer 3 EVPN and Layer 3 VPN over Segment Routing MPLS

This section describes tasks to configure the Layer 3 EVPN and stitching of L3 EVPN and L3VPN router. Perform the following tasks to complete the configuration:

Configuring VRF and Route Targets for Import and Export Rules

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**vrf** *vrf-name*	Defines a VPN routing and forwarding (VRF) instance and enters the VRF configuration mode.
Step 3	**rd auto**	Automatically assigns a unique route distinguisher (RD) to VRF.
Step 4	**address-family** { **ipv4** | **ipv6** } **unicast**	Specifies either the IPv4 or IPv6 address family for the VRF instance and enters address family configuration submode.
Step 5	**route-target import** *route-target-id*	Configures importing of routes to the VRF from the L3VPN BGP NLRIs that have the matching route-target value.

	Command or Action	Purpose
Step 6	**route-target export** *route-target-id*	Configures exporting of routes from the VRF to the L3VPN BGP NLRIs and assigns the specified route-target identifiers to the L3VPN BGP NLRIs.
Step 7	**route-target import** *route-target-id* **evpn**	Configures importing of routes from the L3 EVPN BGP NLRI that have the matching route-target value.
Step 8	**route-target export** *route-target-id* **evpn**	Configures exporting of routes from the VRF to the L3 EVPN BGP NLRIs and assigns the specified route-target identifiers to the BGP EVPN NLRIs.

Configuring BGP EVPN and Label Allocation Mode

You can use MPLS tunnel encapsulation using the **encapsulation mpls** command. You can configure the label allocation mode for the EVPN address family. The default tunnel encapsulation in EVPN for IP Route type in NX-OS is VXLAN.

Advertisement of (IP or Label) bindings from a Cisco Nexus 9000 Series switch via BGP EVPN enables a remote switch to send the routed traffic to that IP using the label for that IP to the switch that advertised the IP over MPLS.

The IP prefix route (Type-5) is:

- Type-5 route with MPLS encapsulation

```
RT-5 Route - IP Prefix

RD:  L3 RD
IP Length:  prefix length
IP address:  IP (4 bytes)
Label1:  BGP MPLS Label
Route Target
RT for IP-VRF
```

The default label allocation mode is per-VRF for Layer 3 EVPN over MPLS.

Complete the following steps to configure BGP EVPN and label allocation mode:

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	Required: **address-family l2vpn evpn**
	Example:
	`switch(config-router)# address-family l2vpn evpn`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the Layer 2 VPN EVPN.

Step 4	Required: **exit**
	Example:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		Exits global address family configuration mode.

Step 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	Example:
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP peer.

Step 6	**address-family l2vpn evpn**
	Example:
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		Advertises the labeled Layer 2 VPN EVPN.

Step 7	**encapsulation mpls**
	Example:
	`router bgp 100`
	`  address-family l2vpn evpn`
	`neighbor NVE2 remote-as 100`
	`  address-family l2vpn evpn`
	`  send-community extended`
	`  encapsulation mpls`
	`  vrf foo`
	`  address-family ipv4 unicast`
	`  advertise l2vpn evpn`
	BGP segment routing configuration:
	`router bgp 100`
	`address-family ipv4 unicast`
		Enables BGP EVPN address family and sends EVPN type-5 route update to the neighbors.
		Note The default tunnel encapsulation in EVPN for the IP route type in NX-OS is VXLAN. To override that, a new CLI is introduced to indicate MPLS tunnel encapsulation.

	Command or Action	Purpose

	`network 200.0.0.1/32 route-map label_index_pol_100`
	`network 192.168.5.1/32 route-map label_index_pol_101`
	`network 101.0.0.0/24 route-map label_index_pol_103`
	`allocate-label all`
	`neighbor 192.168.5.6 remote-as 20`
	`address-family ipv4 labeled-unicast`
	`send-community extended`	

Step 8	**vrf** <*customer_name*>	Configures the VRF.
Step 9	**address-family ipv4 unicast**	Enters global address family configuration mode for the IPv4 address family.
Step 10	**advertise l2vpn evpn**	Advertises Layer 2 VPN EVPN.
Step 11	**redistribute direct route-map DIRECT_TO_BGP**	Redistributes the directly connected routes into BGP-EVPN.
Step 12	**label-allocation-mode per-vrf**	Sets the label allocation mode to per-VRF. If you want to configure the per-prefix label mode, use the **no label-allocation-mode per-vrf** CLI command. For the EVPN address family, the default label allocation is per-vrf, compared to per-prefix mode for the other address-families where the label allocation CLI is supported. No form of CLI is displayed in the running configuration.

Example

See the following example for configuring per-prefix label allocation:

```
router bgp 65000
    [address-family l2vpn evpn]
    neighbor 10.1.1.1
        remote-as 100
        address-family l2vpn evpn
        send-community extended
    neighbor 20.1.1.1
        remote-as 65000
        address-family l2vpn evpn
        encapsulation mpls
        send-community extended
    vrf customer1
        address-family ipv4 unicast
            advertise l2vpn evpn
            redistribute direct route-map DIRECT_TO_BGP
            no label-allocation-mode per-vrf
```

Configuring BGP Layer 3 EVPN and Layer 3 VPN Stitching

In order to configure the stitching on the same router, configure the layer 3 VPN neighbor relationship and router advertisement.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch# configure terminal`
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	Example:
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the Layer 3 VPNv4 or VPNv6.

Step 4	**exit**
	Example:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		Exits global address family configuration mode.

Step 5	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number* **Example:** `switch(config-router)# neighbor 20.1.1.1 remote-as 64498`	Configures the IPv4 address and AS number for a remote BGP L3VPN peer.

Step 6	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	Example:
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		Configure the neighbor address-family for VPNv4 or VPNv6.

Step 7	**send-community extended**	Enables BGP VPN address family

	Command or Action	Purpose
Step 8	**import l2vpn evpn reoriginate**	Configures import of routing information from the Layer 3 VPN BGP NLRIs that has route target identifier matching the normal route target identifier and exports this routing information after re-origination that assigns it with stitching route target identifier, to the BGP EVPN neighbor.

Step 9	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	Example:
	`switch(config-router)# neighbor 10.1.1.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote Layer 3 EVPN BGP peer.

Step 10	**address-family** {**l2vpn** | **evpn**
	Example:
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		Configure the neighbor address-family for Layer 3 EVPN.

Step 11	**import vpn unicast reoriginate**	Enables import of routing information from BGP EVPN NLRIs that has route target identifier matching the stitching route target identifier and exports this routing information after re-origination to the Layer 3 VPN BGP neighbor.
Step 12	**vrf** <*customer_name*>	Configures the VRF.
Step 13	**address-family ipv4 unicast**	Enters global address family configuration mode for the IPv4 address family.
Step 14	**advertise l2vpn evpn**	Advertises Layer 2 VPN EVPN.

Example

```
vrf context Customer1
    rd auto
    address-family ipv4 unicast
        route-target import 100:100
        route-target export  100:100
        route-target import 100:100 evpn
        route-target export 100:100 evpn

segment-routing
    mpls
      global-block 11000 20000
      connected-prefix-sid
         address-family ipv4 unicast
         200.0.0.1 index 101
!
```

```
int lo1
  ip address 200.0.0.1/32
!
interface e1/13
  description “MPLS interface towards Core”
  ip address 192.168.5.1/24
  mpls ip forwarding
  no shut

router bgp 100
address-family ipv4 unicast
allocate-label all
address-family ipv6 unicast
address-family l2vpn evpn
address-family vpnv4 unicast
address-family vpnv6 unicast
 neighbor 10.0.0.1 remote-as 200
    update-source loopback1
    address-family vpnv4 unicast
      send-community extended
      import l2vpn evpn reoriginate
    address-family vpnv6 unicast
      import l2vpn evpn reoriginate
      send-community extended
  neighbor 20.0.0.1 remote-as 300
    address-family l2vpn evpn
      send-community extended
      import vpn unicast reoriginate
      encapsulation mpls
  neighbor 192.168.5.6 remote-as 300
      address-family ipv4 labeled-unicast
  vrf Customer1
    address-family ipv4 unicast
      advertise l2vpn evpn
    address-family ipv6 unicast
      advertise l2vpn evpn
```

Configuring the Features to Enable Layer3 EVPN and Layer3 VPN

Before you begin

Install the VPN Fabric license.

Make sure that the **feature interface-vlan** command is enabled.

Procedure

	Command or Action	Purpose
Step 1	**feature bgp**	Enables BGP feature and configurations.
Step 2	**install feature-set mpls**	Enables MPLS configuration commands.
Step 3	**feature-set mpls**	Enables MPLS configuration commands.
Step 4	**feature mpls segment-routing**	Enables segment routing configuration commands.

	Command or Action	Purpose
Step 5	**feature mpls evpn**	Enables EVPN over MPLS configuration commands. This command is mutually exclusive with the **feature-nv** CLI command.
Step 6	**feature mpls l3vpn**	Enables EVPN over MPLS configuration commands. This command is mutually exclusive with the **feature-nv** CLI command.

Configuring BGP L3 VPN over Segment Routing

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the MPLS segment routing feature.

You must enable the MPLS L3 VPN feature using the **feature mpls l3vpn** command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	**address-family** {**vpnv4** | **vpnv6**} **unicast**
	Example:
	`switch(config-router)# address-family vpnv4 unicast`
	`switch(config-router-af)# address-family vpnv6 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the Layer 3 VPNv4 or VPNv6.

Step 4	[**no**] **allocate-label option-b**	Disables the inter-AS option-b
Step 5	Required: **exit** **Example:**	Exits global address family configuration mode.

	Command or Action	Purpose
	`switch(config-router-af)# exit` `switch(config-router)#`	

Step 6	**neighbor** *ipv4-address* **remote-as** *autonomous-system-number*
	Example:
	`switch(config-router)# neighbor 20.1.1.1 remote-as 64498`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP L3VPN peer.

Step 7	**address-family** {**vpnv4** | **vpnv6** } **unicast**
	Example:
	`switch(config-router-neighbor)# address-family vpnv4 unicast`
	`switch(config-router-neighbor-af)#`
		Configure the neighbor address-family for VPNv4 or VPNv6.

Step 8	**send-community extended**	Enables BGP VPN address family.
Step 9	**vrf** <*customer_name*>	Configures the VRF.
Step 10	**allocate-index x**	Configure the allocate-index.
Step 11	**address-family ipv4 unicast**	Enters global address family configuration mode for the IPv4 address family.
Step 12	**redistribute direct route-map DIRECT_TO_BGP**	Redistributes the directly connected routes into BGP-L3VPN.

BGP Layer3 VPN Over SRTE

This feature enables the traffic engineering capabilities towards the Segment Routing core for Data-Center Interconnect (DCI)/WAN Edge deployments. It enables DCI hand off (VxLAN to L3VPN based on SR and vice-versa) and can use SRTE capabilities in SR Core so that SLA's can be achieved by different traffic classes. SRTE capabilities can be applied on DCI or edge routers by applying SR-Policy for L3VPN prefixes. L3VPN prefixes can be advertised(by DCI or Edge nodes) after setting extended community color and BGP L3VPN neighbor can apply SR-policy based on that color to create SRTE. Listed below are the configurations for configuring extended community color on L3VPN prefixes.

Guidelines and Limitations for Configuring Layer 3 VPN Over SRTE

Beginning with Cisco NX-OS Release 10.1(2), segment routing traffic engineering is supported over Layer 3 VPN on Cisco Nexus 9300-FX3, N9K-C9316D-GX , N9K-C93180YC-FX, N9K-C93240YC-FX2, and N9K-C9364C platform switches.

The limitations for this feature are as follows:

- UnderLay IPv6 is not supported. SRv6 is the alternate.
- PCE using BGP underlay is not supported, due to PCE's shortcoming on BGP only fabric.
- OSPF-SRTE with PCE is not supported, due NXOS's inability to advertise LSA in BGP-LS.

- Supports total SRTE policy scale of 1000, BGP VPNv4 32K routes, BGP VPNV6 32k routes, and underlay SR prefixes of 1000.

Beginning with Cisco NX-OS Release 10.2(3)F, the option of color-only (CO) bits is added in route map. If the value of the CO bits change for a given prefix that is using an SRTE policy, BGP will delete the old policy and add a new policy.

Configuring Extended Community Color

This section includes the following topics:

Configuring Extended Community Color at the Ingress Node

To configure extended community color at the ingress node when the prefix is announced by the ingress node, where the SRTE policy is instantiated, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	**set extcommunity color** *color-num*
	Example:
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		Sets BGP extcommunity attribute for color extended community.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

	Command or Action	Purpose

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family vpnv4/vpnv6 unicast**
	Example:
	`switch(config-router-neighbor)# address-family vpnv4/vpnv6 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address-family configuration mode for the vpnv4/vpnv6 address family type.

Step 8	**route-map** *map-name* **in**
	Example:
	`switch(config-router-neighbor-af)# route-map ABC in`
	`switch(config-router-neighbor-af)#`
		Applies the configured BGP policy to incoming routes.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Extended Community Color at the Egress Node

To configure extended community color at the egress node when the prefix is announced by the egress node, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	**set extcommunity color** *color-num*
	Example:
	`switch(config-route-map)# set extcommunity color 20`
	`switch(config-route-map)#`
		Sets BGP extcommunity attribute for color extended community.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

	Command or Action	Purpose

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family vpnv4/vpnv6 unicast**
	Example:
	`switch(config-router-neighbor)# address-family vpnv4/vpnv6 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address-family configuration mode for the vpnv4/vpnv6 address family type.

Step 8	**route-map** *map-name* **out**
	Example:
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		Applies the configured BGP policy to outgoing routes.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Extended Community Color for Network/Redistribute Command at the Egress Node

To configure extended community color for the network/redistribute command at the egress node when the prefix is announced by the egress node, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	**set extcommunity color** *color-num* **Example:**	Sets BGP extcommunity attribute for color extended community.

	Command or Action	Purpose
	`switch(config-route-map)# set extcommunity color 20` `switch(config-route-map)#`	

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1;`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**vrf** <*customer_name*>	Configures the VRF.

Step 7	**address-family ipv4 unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Specifies the IPv4 address family for the VRF instance and enters the address family configuration mode.

Step 8	**redistribute static route-map** *map-name* **out**
	Example:
	`switch(config-router-vrf-af)# redistribute static route-map ABC`
	`switch(config-router-af)#`
		Redistributes static routes into BGP. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Step 9	**network** *ip-prefix* [**route-map** *map-name*]
	Example:
	`switch(config-router-vrf-af)# network 1.1.1.1/32 route-map ABC`
	`switch(config-router-af-network)#`
		Specifies a network as local to this autonomous system and adds it to the BGP routing table.

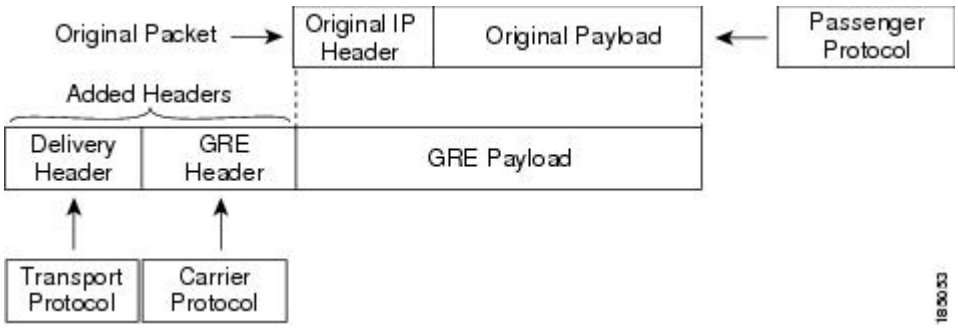
Configuring Segment Routing MPLS and GRE Tunnels

GRE Tunnels

You can use generic routing encapsulation (GRE) as the carrier protocol for a variety of passenger protocols.

The following figure shows the IP tunnel components for a GRE tunnel. The original passenger protocol packet becomes the GRE payload and the device adds a GRE header to the packet. The device then adds the transport protocol header to the packet and transmits it.

Figure 5: GRE PDU



Segment Routing MPLS and GRE

Beginning Cisco NX-OS Release 9.3(1), you can configure both, segment routing MPLS and generic routing encapsulation (GRE) on a Cisco Nexus device. Both these technologies operate seamlessly. All MPLS traffic can be forwarded to the GRE tunnel after the MPLS tunnel termination. Similarly, you can forward all traffic from the GRE tunnel to the MPLS cloud after the GRE termination.

All PE routers can initiate, forward, or terminate the GRE traffic from or to another GRE cloud. Similarly, all tunnel transit or tunnel end nodes can configure MPLS tunnel encapsulation.

When both, the tunnel and segment routing is enabled on the Cisco Nexus 9000 switches, the following is the TTL behavior is for the respective flows:

- Incoming IP traffic, egresses with GRE header, the TTL value in the GRE header is one less than the TTL value of the incoming IP packet.
- Incoming IP traffic, egresses with MPLS header, the TTL value in the MPLS header is one less than the TTL value of the incoming IP packet.
- Incoming GRE traffic, egresses with MPLS header, the TTL value in the MPLS header is default (255).
- Incoming MPLS traffic, egresses with GRE header, the TTL value in the GRE header is default (255).

Guidelines and Limitations for Segment Routing MPLS and GRE

Segment routing MPLS and GRE have the following guidelines and limitations:

- Ingress stats are not supported for tunnel packets.
- Only template-mpls-heavy template is supported.
- MPLS segment routing is not supported on the tunnel interfaces.
- Due to a hardware limitation on the modular switches, the tunnel Tx traffic is not supported if the egress interface for the tunnel destination IP address is over the Cisco Nexus 9300-FX/FX2 platform switches.
- Maximum four GRE tunnels are supported.
- Beginning with Cisco NX-OS Release 9.3(3), you can configure both, segment routing MPLS and GRE on Cisco Nexus 9300-GX platform switches.
- Tunnel Rx packet counters do not work when both segment routing MPLS and GRE coexist.

Configuring Segment Routing MPLS and GRE

You can enable MPLS segment routing as long as mutually-exclusive MPLS features such as static MPLS are not enabled.

Before you begin

You must install and enable the MPLS feature set using the **install feature-set mpls** and **feature-set mpls** commands.

You must enable the tunneling feature using the **feature tunnel** command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature segment-routing** **Example:** `switch(config)# feature segment-routing`	Enables the MPLS segment routing feature. The **no** form of this command disables the MPLS segment routing feature.
Step 3	(Optional) **show running-config** | **inc 'feature segment-routing'** **Example:** `switch(config)# show running-config | inc 'feature segment-routing'`	Displays the status of the MPLS segment routing feature.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Step 5	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 6	**feature tunnel**
	Example:
	`switch(config)# feature tunnel`
	`switch(config-if)#`
		Allows the creation of a new tunnel interface.
		To disable the tunnel interface feature, use the **no** form of this command.

Step 7	switch(config)# **interface tunnel** *number*	Enters a tunnel interface configuration mode.
Step 8	switch(config-if)# **tunnel mode** {**gre ip** }	Sets this tunnel mode to GRE.

	Command or Action	Purpose
		The **gre** and **ip** keywords specify that GRE encapsulation over IP will be used.
Step 9	**tunnel source** {*ip-address* |*interface-name*} **Example:** `switch(config-if)# tunnel source ethernet 1/2`	Configures the source address for this IP tunnel. The source can be specified by IP address or logical interface name.
Step 10	**tunnel destination** {*ip-address* |*host-name*} **Example:** `switch(config-if)# tunnel destination 192.0.2.1`	Configures the destination address for this IP tunnel. The destination can be specified by IP address or logical host name.
Step 11	**tunnel use-vrf** *vrf-name* **Example:** `switch(config-if)# tunnel use-vrf blue`	
Step 12	**ipv6 address** *IPv6 address*	`switch(config-if)# 10.1.1.1` Configures the IPv6 address. **Note** The tunnel source and the destination addresses are still the same (IPv4 address.)
Step 13	(Optional) switch(config-if)# **show interface tunnel** *number*	Displays the tunnel interface statistics.
Step 14	switch(config-if)# **mtu** *value*	Sets the maximum transmission unit (MTU) of IP packets sent on an interface.
Step 15	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Verifying the Segment Routing MPLS and GRE Configuration

To display the segment routing MPLS and GRE configuration, perform one of the following tasks:

Command	Purpose
show segment-routing mpls	Displays segment routing mpls information

Verifying SR-TE for Layer 3 EVPN

The ODN verifications are based on L3VPN VRF prefixes.

1. Verify that the PCEP session between R1 (headend and PCE server) is established.

```
R1# show srte pce ipv4 peer

PCC's peer database:
--------------------
Remote PCEP conn IPv4 addr: 58.8.8.8
Local PCEP conn IPv4 addr: 51.1.1.1
Precedence: 0
State: up
```

2. Verify BGP LS and BGP EVPN session on R1, R3, and R6 using the following commands:
   - Show bgp l2vpn evpn summary
   - Show bgp link-state summary
3. Verify that the R1 (headend) has no visibility to the R6 loopback address.

```
R1# show ip route 56.6.6.6
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

56.6.6.6/32, ubest/mbest: 1/0
    *via Null0, [1/0], 1d02h, static
```

4. Verify that the VRF prefix is injected via MP-BGP in a R1 VRF SR routing table.

```
R1# show ip route vrf sr
106.107.4.1/32, ubest/mbest: 1/0
    *via binding label 100534%default, [20/0], 1d01h, bgp-6503, external, tag 6500
(mpls-vpn)
```

5. Verify the SR-TE Tunnel.

```
R1# show srte policy
Policy name: 51.1.1.1|1001
    Source: 51.1.1.1
    End-point: 56.6.6.6
    Created by: bgp
    State: UP
    Color: 1001
    Insert: FALSE
    Re-opt timer: 0
    Binding-sid Label: 100534
    Policy-Id: 2
    Flags:
    Path type = MPLS           Path options count: 1
     Path-option Preference:100 ECMP path count: 1
      1.      PCE         Weighted: No
        Delegated PCE: 58.8.8.8
                Index: 1                Label: 101104
                Index: 2                Label: 201102
                Index: 3                Label: 201103
```

Verifying the Segment Routing Configuration

To display the segment routing configuration, perform one of the following tasks:

Command	Purpose
show bgp ipv4 labeled-unicast *prefix*	Displays the advertised label index and the selected local label for the specified IPv4 prefix.
show bgp paths	Displays the BGP path information, including the advertised label index.
show mpls label range	Displays the configured SRGB range of labels.
show route-map [*map-name*]	Displays information about a route map, including the label index.
show running-config rpm	Displays information about Route Policy Manager (RPM).
show running-config | **inc 'feature segment-routing'**	Displays the status of the MPLS segment routing feature.
show ip ospf neighbors detail	Displays the list of OSPFv2 neighbors and the adjacency SID allocated, along with the corresponding flags.
show ip ospf database opaque-area	Displays the LSAs for the adjacency SID.
show ip ospf segment-routing adj-sid-database	Displays all locally allocated adjacency SIDs.
show running-config segment-routing	Displays the status of the segment routing feature.
show srte policy	Displays only the authorized policies.
show srte policy [**all**]	Displays the list of all policies available in the SR-TE.
show srte policy [**detail**]	Displays the detailed view of all the requested policies.
show srte policy <*name*>	Filters the SR-TE policy with the name and displays the list of all policies available with that name in the SR-TE. **Note** This command has the autocomplete feature for the policy-name. To use this feature, add a question mark or press TAB.
show srte policy color <*color*> **endpoint** <*endpoint*>	Displays the SR-TE policy for the color and endpoint. **Note** This command has the autocomplete feature for color and endpoint. To use this feature, add a question mark or press TAB.
show srte policy fh	Displays the set of first hops.
show segment-routing mpls clients	Displays the clients registered with the SR-APP.

Command	Purpose
show segment-routing mpls details	Displays detailed information.
show segment-routing ipv4	Displays the information for the IPv4 address family.
show segment-routing mpls	Displays segment routing mpls information
show segment-routing ipv4 connected-prefix-sid	Displays the MPLS label range for the SRGB. **Note** This command is only available in Cisco NX-OS Release 9.3(1) .
show ip ospf *process*	Displays the OSPF mode.
show ip ospf *process* **segment-routing sid-database**	Displays the segment routing database details.
show ip ospf *process* **segment-routing global block**	Displays the segment routing global block information.
show nve evi	Displays the status of the EVIs.
show nve peer mpls	Displays the status of the segment routing peers.
show nve adjacency mpls	Displays the status of the peer adjacencies.

Configuring SRTE Explicit-Path Endpoint Substitution

This chapter contains information on how to configure the SRTE Explicit-path Endpoint Substitution feature.

About SRTE Explicit-path Endpoint Substitution

The SRTE Explicit-path Endpoint Substitution feature allows the user to define an explicit path as a series of MPLS labels, like a regular explicit path, but allows a placeholder to be added in the series that represents the policy endpoint label. The placeholder is represented by the **policy-endpoint** keyword. The position in the path where the policy-endpoint placeholder appears is resolved by SRTE internally to the Segment Routing label representing the node SID of the endpoint IP address of the policy.

This is valuable when used in conjunction with on-demand color templates since it reduces the total number of policies that must be defined. Rather than define a separate path for each color and endpoint combination, instead the user can define an on-demand color template that contains an explicit path with endpoint substitution to define policies for all endpoints of that color.

Guidelines and Limitations for SRTE Explicit-path Endpoint Substitution

SRTE Explicit-path Endpoint Substitution has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.1(1), SRTE Explicit-path Endpoint Substitution is supported on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, and 9300-GX platform switches.
- If the partial path ends in the same label as the resolved endpoint label, do not append the extra (duplicated) transport label.

- SRGB must be the same on all nodes; if not, the feature may not work depending on the segment configuration of each intermediate node.
- A segment list can have only one policy-endpoint entry.

Configuring SRTE Explicit-path Endpoint Substitution

To create a policy that uses endpoint substitution, first define the path using the segment-list mode. Then associate the path with an on-demand color using its name.

Before you begin

You must ensure that the MPLS segment routing traffic engineering feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enters the segment routing configuration mode.

Step 3	**traffic-engineering**
	Example:
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		Enters the traffic engineering mode.

Step 4	**segment-list name** *path*
	Example:
	`switch(config-sr-te)# segment-list name path`
	`switch(config-sr-te-exp-seg-list)#`
		Configures an explicit segment list.

Step 5	**index 1 mpls label** *label-ID*
	Example:
	`switch(config-sr-te-exp-seg-list)# index 1 mpls label 16201`
	`switch(config-sr-te-exp-seg-list)#`
		Configures an MPLS label in the segment list.

Step 6	**index 2 policy-endpoint**
	Example:
	`switch(config-sr-te-exp-seg-list)# index 2 policy-endpoint`
	`switch(config-sr-te-exp-seg-list)#`
		Configures the policy endpoint resolution.

	Command or Action	Purpose

Step 7	**exit**
	Example:
	`switch(config-sr-te-exp-seg-list)# exit`
	`switch(config-sr-te)#`
		Exits the segment list mode and returns to the SRTE mode.

Step 8	**on-demand color** *color_num*
	Example:
	`switch(config-sr-te)# on-demand color 201`
	`switch(config-sr-te-color)#`
		Enters the on-demand color template mode to configure an on-demand color for the specified color.

Step 9	**candidate-paths** **Example:** `switch(config-sr-te-color)# candidate-paths`	Specifies the candidate paths for the SR-TE color policy.
Step 10	**preference** *preference-number* **Example:** `switch(cfg-cndpath)# preference 100`	Specifies the preference of the candidate path.
Step 11	**explicit segment-list** *path* **Example:** `switch(cfg-pref)# explicit segment-list path`	Specifies the explicit segment list.

Configuration Example for SRTE Explicit-path Endpoint Substitution

This example shows the SRTE Explicit-path Endpoint Substitution configuration:

```
switch(config)# segment-routing
switch(config-sr)# traffic-engineering
switch(config-sr-te)# segment-list name path
switch(config-sr-te-exp-seg-list)# index 1 mpls label 16201
switch(config-sr-te-exp-seg-list)# index 2 policy-endpoint
switch(config-sr-te-exp-seg-list)# exit
switch(config-sr-te)# on-demand color 201
switch(config-sr-te-color)# candidate-paths
switch(cfg-cndpath)# preference 100
switch(cfg-pref)# explicit segment-list path
```

Verifying Configuration for SRTE Explicit-path Endpoint Substitution

To display the required details about the SRTE Explicit-path Endpoint Substitution configuration, perform one of the following tasks:

Table 3: Verifying the SRTE Explicit-path Endpoint Substitution Configuration

Command	Purpose
show srte policy	Displays only the authorized policies. **Note** If the endpoint label is resolved and the first hop is reachable, the state is displayed as UP. If the endpoint label is not resolved or the first hop is not reachable, the state is displayed as DOWN.
show srte policy [**all**]	Displays the list of all policies available in the SR-TE. **Note** If the endpoint label is resolved and the first hop is reachable, the state is displayed as UP. If the endpoint label is not resolved or the first hop is not reachable, the state is displayed as DOWN.
show srte policy [**detail**]	Displays the detailed view of all the requested policies. **Note** If the endpoint label is resolved and the first hop is reachable, the state is displayed as UP. If the endpoint label is not resolved or the first hop is not reachable, the state is displayed as DOWN.
show srte policy <*name*>	Filters the SR-TE policy with the name and displays the list of all policies available with that name in the SR-TE. **Note** This command has the autocomplete feature for the policy-name. To use this feature, add a question mark or press TAB.
show srte policy color <*color*> **endpoint** <*endpoint*>	Displays the SR-TE policy for the color and endpoint. **Note** This command has the autocomplete feature for color and endpoint. To use this feature, add a question mark or press TAB.
show srte policy fh	Displays the state of the existing first hop and policy endpoints.

Configuring SRTE Over Default VRF

About SRTE Over Default VRF

The SRTE Over Default VRF feature allows you to incorporate segment routing traffic engineering to achieve the traffic steering benefits in your network. The SRTE provides increased scalability while using BGP for routing in large-scale data centers (DC).

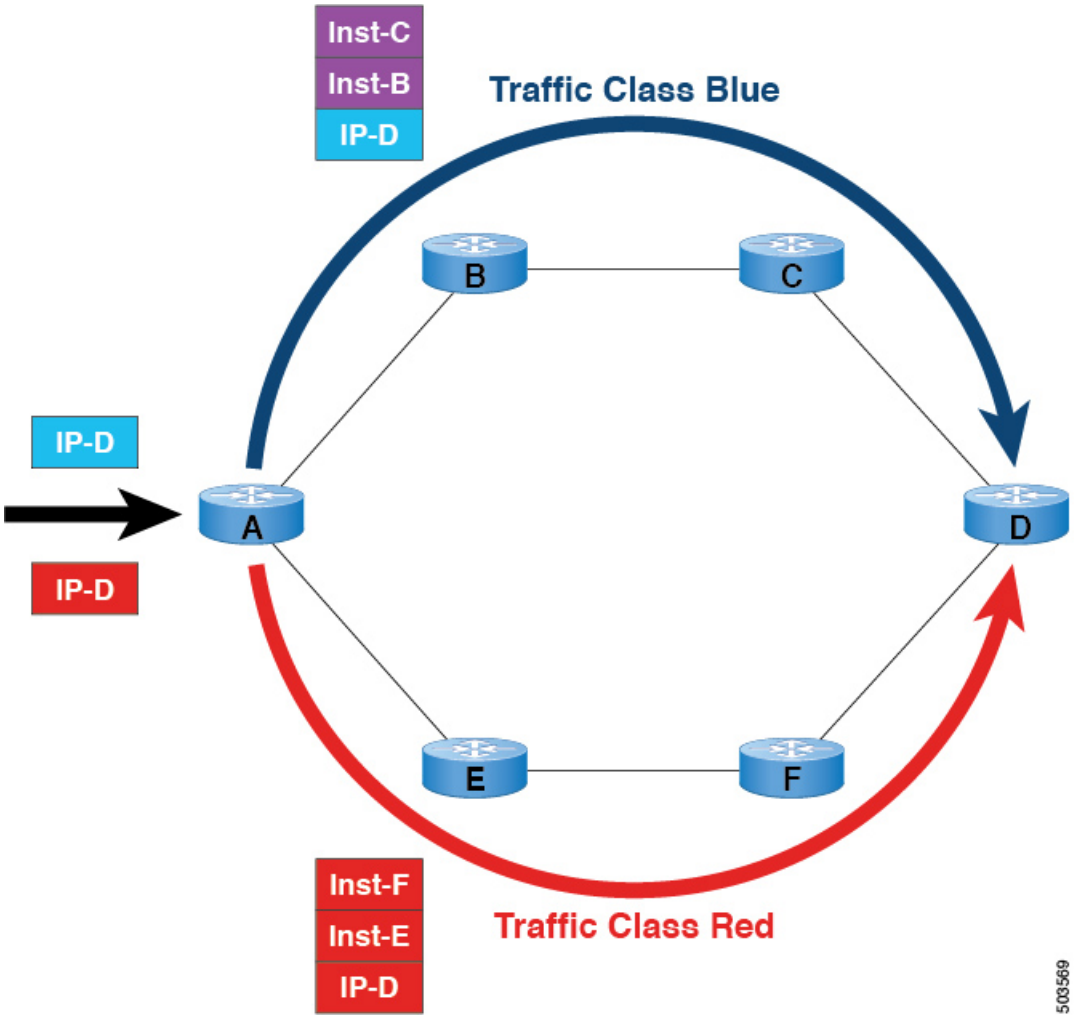
The SRTE Over Default VRF feature uses the route color that exists as an extended community attribute and is represented by a number as the base for traffic steering. Based on the color, plane separation is achieved, and an SR policy is created to carry the traffic. Furthermore, based on the color, the DC is divided into different planes. The applications are configured to use each plane to only route through a specific plane and steer traffic to appropriate destinations.

Plane separation has the following advantages:

- One flow does not affect the other flow.
- Large and small flows are separated into different planes.
- Fault isolation for better debuggability: Fault in one plane does not affect the other planes. For example, if a network fault occurs in one plane, only the applications in that plane are affected, but the applications in the rest of the planes are not impacted. Additionally, the fault can be isolated and troubleshooted in isolation.

The following example explains the SRTE Over Default VRF feature with an illustration.

Figure 6: SRTE Over Default VRF Example



- For BGP, node A is the ingress router and node D is the egress router. D is also the next-hop.
- For SRTE, node A is the SRTE headend, node D is the endpoint for the policy.
- Route prefix 1 is configured to use the blue plane, and route 2 is configured to use the red plane.

The blue traffic is appended with instructions to steer the traffic through node B and node C, and the red traffic is appended with instruction to steer traffic through node E and node F. In summary, the traffic is handled based on the color of the advertisement, that is, the prefix that was advertised earlier.

Guidelines and Limitations for Configuring SRTE Over Default VRF

- Beginning with Cisco NX-OS Release 10.1(1), segment routing traffic engineering is supported over default VRF on Cisco Nexus 9300-FX3, N9K-C9316D-GX , N9K-C93180YC-FX, N9K-C93240YC-FX2, and N9K-C9364C platform switches. The limitations for this SR-TE feature are as follows:
  - UnderLay IPv6 is not supported. SRv6 is the alternate.

- PCE using BGP underlay is not supported, due to PCE's shortcoming on BGP only fabric.
- OSPF-SRTE with PCE is not supported, due to NXOS' inability to advertise LSA in BGP-LS.
- Supports total SRTE policy scale of 1000, BGP Default VRF(v4) of 130K v4, and underlay SR prefixes of 1000.

- Beginning with Cisco NX-OS Release 10.2(3)F, the option of color-only (CO) bits is added in route map. If the value of the CO bits change for a given prefix that is using an SRTE policy, BGP will delete the old policy and add a new policy. This feature is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX, and 9300-GX2 platform switches./

Configuration Process: SRTE Over Default VRF

The configuration process is as follows:

1. Set next-hop unchanged: The next-hop is used to calculate the SR policy at the ingress node. The next-hop in the SR domain on a prefix must be preserved as the prefix is advertised upstream. Hence, next-hop unchanged in needed on all upstream routers in the case for hop-by-hop ebgp.
2. Set extended community color at the egress node, ingress node, network/redistribute, or default-originate.
3. The ingress node, on receiving a color-extended community, matches it to an SR policy.
4. The endpoint for the SR policy is derived from the next-hop of the prefix and color in the color-extended community.

This section includes the following topics on configuring SRTE over default VRF:

Configuring Next-hop Unchanged

To configure next-hop unchanged on the intermediate (spine) nodes for default VRF overlay, to ensure the next-hop is not changed, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	[**no**] **set ip next-hop unchanged** **Example:**	Sets next-hop unchanged.

	Command or Action	Purpose
	`switch(config-route-map)# set ip next-hop unchanged` `switch(config-route-map)#`	

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family ipv4 unicast**
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address-family configuration mode for the IPv4 address family type.

Step 8	**route-map** *map-name* **out**
	Example:
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		Applies the configured BGP policy to outgoing routes.

Configuring Extended Community Color

This section includes the following topics:

Configuring Extended Community Color at the Egress Node

To configure extended community color at the egress node when the prefix is announced by the egress node, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	Example:
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		Sets BGP extcommunity attribute for color extended community.
		co-flag: Use the color-only flag to control whether traffic may be steered into an SR Policy based on color only, if no policy can be found for the exact color and endpoint. The default is 00.
		Note Select the co-flag 00 to specify the default Automated Steering function based on color and nexthop. When the co-flag is 00 or set to default, the binding sid of the policy with the requested color and endpoint is used for routing.
		Select the co-flag 01 to steer traffic based on color only. When the co-flag is set to 01, and if the policy with requested color and endpoint exists, the binding sid of the policy is used for routing. If the policy does not exist, but the null endpoint policy with the same color exists, then the binding sid of the null endpoint policy is used for routing.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number* **Example:**	Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of

	Command or Action	Purpose
	`switch(config)# router bgp1` `switch(config-router)#`	a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family ipv4 unicast**
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address-family configuration mode for the IPv4 address family type.

Step 8	**route-map** *map-name* **out**
	Example:
	`switch(config-router-neighbor-af)# route-map ABC out`
	`switch(config-router-neighbor-af)#`
		Applies the configured BGP policy to outgoing routes.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Extended Community Color at the Ingress Node

To configure extended community color at the ingress node when the prefix is announced by the ingress node, where the SRTE policy is instantiated, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	Example:
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		Sets BGP extcommunity attribute for color extended community.
		co-flag: Use the color-only flag to control whether traffic may be steered into an SR Policy based on color only, if no policy can be found

	Command or Action	Purpose
		for the exact color and endpoint. The default is 00. **Note** Select the co-flag 00 to specify the default Automated Steering function based on color and nexthop. When the co-flag is 00 or set to default, the binding sid of the policy with the requested color and endpoint is used for routing. Select the co-flag 01 to steer traffic based on color only. When the co-flag is set to 01, and if the policy with requested color and endpoint exists, the binding sid of the policy is used for routing. If the policy does not exist, but the null endpoint policy with the same color exists, then the binding sid of the null endpoint policy is used for routing.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family ipv4 unicast**
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address-family configuration mode for the IPv4 address family type.

	Command or Action	Purpose

Step 8	**route-map** *map-name* **in**
	Example:
	`switch(config-router-neighbor-af)# route-map ABC in`
	`switch(config-router-neighbor-af)#`
		Applies the configured BGP policy to incoming routes.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Extended Community Color for Network/Redistribute Command at the Egress Node

To configure extended community color for the network/redistribute command at the egress node when the prefix is announced by the egress node, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.

Step 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*]
	Example:
	`switch(config-route-map)# set extcommunity color 20 [co-flag 00]`
	`switch(config-route-map)#`
		Sets BGP extcommunity attribute for color extended community.
		co-flag: Use the color-only flag to control whether traffic may be steered into an SR Policy based on color only, if no policy can be found for the exact color and endpoint. The default is 00.

	Command or Action	Purpose
		Note Select the co-flag 00 to specify the default Automated Steering function based on color and nexthop. When the co-flag is 00 or set to default, the binding sid of the policy with the requested color and endpoint is used for routing. Select the co-flag 01 to steer traffic based on color only. When the co-flag is set to 01, and if the policy with requested color and endpoint exists, the binding sid of the policy is used for routing. If the policy does not exist, but the null endpoint policy with the same color exists, then the binding sid of the null endpoint policy is used for routing.

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Specifies the IPv4 address family for the VRF instance and enters the address family configuration mode.

Step 7	**redistribute static route-map** *map-name* **out**
	Example:
	`switch(config-router-af)# redistribute static route-map ABC`
	`switch(config-router-af)#`
		Redistributes static routes into BGP. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Step 8	**network** *ip-prefix* [**route-map** *map-name*] **Example:**	Specifies a network as local to this autonomous system and adds it to the BGP routing table.

	Command or Action	Purpose
	`switch(config-router-af)# network 1.1.1.1/32 route-map ABC` `switch(config-router-af-network)#`	

Configuring Extended Community Color for Default-Originate at the Egress Node

To configure extended community color for default-originate at the egress node when the default prefix is announced by the egress node, perform the following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**route-map** *map-name*
	Example:
	`switch(config)# route-map ABC`
	`switch(config-route-map)`
		Creates a route map or enters route-map configuration mode for an existing route map.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Step 3	**set extcommunity color** *color-num* [**co-flag** *co-flag*] **Example:** `switch(config-route-map)# set extcommunity color 20 [co-flag 00]`	Sets BGP extcommunity attribute for color extended community. **co-flag:** Use the color-only flag to control whether traffic may be steered into an SR Policy based on color only, if no policy can be found for the exact color and endpoint. The default is 00. **Note** Select the co-flag 00 to specify the default Automated Steering function based on color and nexthop. When the co-flag is 00 or set to default, the binding sid of the policy with the requested color and endpoint is used for routing. Select the co-flag 01 to steer traffic based on color only. When the co-flag is set to 01, and if the policy with requested color and endpoint exists, the binding sid of the policy is used for routing. If the policy does not exist, but the null endpoint policy with the same color exists, then the binding sid of the null endpoint policy is used for routing.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits route-map configuration mode.

Step 5	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp1`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 6	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 7	**address-family ipv4 unicast**
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address-family configuration mode for the IPv4 address family type.

Step 8	**default-originate** [**route-map** *map-name*]
	Example:
	`switch(config-router-neighbor-af)# default-originate route-map ABC`
	`switch(config-router-neighbor-af)#`
		Generates a default route to the BGP peer.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring BGP for Ingress Peer (SRTE Headend)

To configure BGP for the ingress peer (SRTE headend), perform the following steps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature bgp** **Example:**	Enables BGP. Use the no form of this command to disable this feature.

	Command or Action	Purpose
	`switch(config)# feature bgp` `switch(config)`	

Step 3	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 4	**address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

Step 5	**neighbor** *ip-address*
	Example:
	`switch(config-router-af)# neighbor 209.165.201.1`
	`switch(config-router-af-neighbor)#`
		Configures the IPv4 address for a remote BGP peer. The ip-address format is x.x.x.x.

Step 6	**remote-as** *as-number* **Example:** `switch(config-router-af-neighbor)# remote-as 64497`	Configures the AS number for a remote BGP peer.
Step 7	**update-source** *interface number* **Example:** `switch(config-router-af-neighbor)# update-source loopback 300`	Specifies and updates the source of the BGP session.
Step 8	**ebgp-multihop** *ttl-value* **Example:** `switch(config-router-af-neighbor)# ebgp-multihop 5`	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.
Step 9	**exit** **Example:** `switch(config-router-af-neighbor)# exit`	Exits the neighbor configuration mode.

Step 10	**address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

	Command or Action	Purpose
Step 11	**route-map** *map-name* **in** **Example:** `switch(config-router-af)# route-map color 401 in`	Specifies the route map for the SRTE ingress peer. The map-name can be any case-sensitive, alphanumeric string up to 63 characters. **Note** Only one extended community color can be applied to an NLRI, so any route-policy/route-map applied overrides the previous extended community color, if it exists.

Configuring BGP for Egress Peer (SRTE Endpoint)

To configure BGP for the egress peer (SRTE endpoint), perform the following steps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)`
		Enables BGP.
		Use the no form of this command to disable this feature.

Step 3	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the no option with this command to remove the BGP process and the associated configuration.

Step 4	**neighbor** *ip-address*
	Example:
	`switch(config-router)# neighbor 209.165.201.1`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address for a remote BGP peer. The ip-address format is x.x.x.x.

Step 5	**remote-as** *as-number* **Example:**	Configures the AS number for a remote BGP peer.

	Command or Action	Purpose
	`switch(config-router-neighbor)# remote-as 64497`	
Step 6	**update-source** *interface-number* **Example:** `switch(config-router-neighbor)# update-source loopback 300`	Specifies and updates the source of the BGP session.
Step 7	**ebgp-multihop** *ttl-value* **Example:** `switch(config-router-neighbor)# ebgp-multihop 5`	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.
Step 8	**exit** **Example:** `switch(config-router-af-neighbor)# exit`	Exits the neighbor configuration mode.

Step 9	**address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.

Step 10	**send-community**
	Example:
	`switch(config-router-af)# send-community`
	`switch(config-router-af)#`
		Specifies that the BGP community attribute must be sent to a BGP neighbor.

Step 11	**send-community extended**
	Example:
	`switch(config-router- af)#send-community extended`
	`switch(config-router-af)#`
		Specifies that extended communities attribute should be sent to a BGP neighbor.

Step 12	**route-map** *map-name* **out**
	Example:
	`switch(config-router-af)# route-map color 301 out`
	`switch(config-router-af)#`
		Specifies the route map for the SRTE egress peer.
		The map-name can be any case-sensitive, alphanumeric string up to 63 characters.
		Note Only one extended community color can be applied to an NLRI, so any route-policy/route-map applied overrides the previous extended community color, if it exists.

Configuring SRTE for Ingress Peer (SRTE Headend)

To configure the SRTE for ingress peer (SRTE headend), perform the following steps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature mpls segment-routing traffic-engineering**
	Example:
	`switch(config)# feature mpls segment-routing traffic-engineering`
	`switch(config)`
		Enables MPLS SRTE.
		Use the no form of this command to disable this feature.

Step 3	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enters the segment routing configuration mode.

Step 4	**traffic-engineering**
	Example:
	`switch(config-sr)# traffic-engineering`
	`switch(config-sr-te)#`
		Enters the traffic engineering mode.

Step 5	**segment-list name** *path*
	Example:
	`switch(config-sr-te)# segment-list name path`
	`switch(config-sr-te-exp-seg-list)#`
		Configures an explicit segment list.

Step 6	**index 1 mpls label** *label-ID*
	Example:
	`switch(config-sr-te-exp-seg-list)# index 1 mpls label 16601`
	`switch(config-sr-te-exp-seg-list)#`
		Create an MPLS label in the segment list.

Step 7	**index 2 mpls label** *label-ID*
	Example:
	`switch(config-sr-te-exp-seg-list)# index 2 mpls label 16501`
	`switch(config-sr-te-exp-seg-list)#`
		Creates MPLS label in the segment list.

Step 8	**policy** *policy-name*-**bgp** **Example:**	Specifies the SRTE policy name.

	Command or Action	Purpose
	`switch(config-sr-te-exp-seg-list)# policy dci1-edge1-bgp` `switch(config-sr-te-exp-seg-list)#`	
Step 9	**color** *color-num* **endpoint** *endpoint ID* **Example:** `switch(config-sr-te)# color 13401 endpoint 1.0.3.1`	Specifies the color and endpoint for the policy (SRTE Egress Node Loopback).
Step 10	**candidate-paths** **Example:** `switch(config-sr-te-color)# candidate-paths`	Specifies the candidate paths for the SRTE color policy.
Step 11	**preference** *preference-number* **Example:** `switch(cfg-cndpath)# preference 100`	Specifies the preference of the candidate path.
Step 12	**explicit segment-list** *path* **Example:** `switch(cfg-pref)# explicit segment-list path`	Specifies the explicit segment list.

Configuration Example for SRTE Over Default VRF

The following examples show the SRTE over default VRF configuration:

Configuration Example: Next-hop Unchanged

```
route-map ABC
  set ip next-hop unchanged

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC out
```

Configuration Examples: Extended Community Color

This section includes the following configuration examples for extended community color:

Configuration Example: At the Egress Node

```
ip prefix-list pfx1 seq 5 permit 7.7.7.7/32
ip prefix-list pfx2 seq 5 permit 5.0.0.0/24
route-map ABC
  match ip address prefix-list pfx1 pfx2
    set extcommunity color 20

router bgp 1
  neighbor 1.2.3.4
```

```
    address-family ipv4 unicast
       route-map ABC out
```

Configuration Example: At the Ingress Node

```
ip prefix-list pfx1 seq 5 permit 7.7.7.7/32
ip prefix-list pfx2 seq 5 permit 5.0.0.0/24
route-map ABC
  match ip address prefix-list pfx1 pfx2
    set extcommunity color 20

router bgp 1
  neighbor 1.2.3.4
    address-family ipv4 unicast
       route-map ABC in
```

Configuration Example: For Network/Redistribute Command at the Egress Node

```
route-map ABC
  set extcommunity color 20

router bgp 1
  address-family ipv4 unicast
    redistribute static route-map ABC
    network 1.1.1.1/32 route-map ABC
```

Configuration Example: For Default-Originate at the Egress Node

```
route-map ABC
  set extcommunity color 20

router bgp 1
    neighbor 1.2.3.4
       address-family ipv4 unicast
           default-originate route-map ABC
```

Configuration Example: BGP for Ingress Peer (SRTE Headend)

```
DCI-1(config)# show running-config bgp
feature bgp
router bgp 100
  address-family ipv4 unicast
  neighbor 1.0.3.1
    remote-as 101
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
       route-map color-3401 in
```

Configuration Example: BGP for Egress Peer (SRTE Endpoint)

This example shows the SRTE Explicit-Path Endpoint Substitution configuration:

```
Edge-1(config)# show running-config bgp
feature bgp
router bgp 101
neighbor 1.0.1.1
    remote-as 100
    update-source loopback0
    ebgp-multihop 255
    address-family ipv4 unicast
      send-community
```

```
    send-community extended
    route-map color-3401 out
```

Configuration Example: Ingress Peer for SRTE (SRTE Headend)

```
DCI-1# show running-config srte
feature mpls segment-routing traffic-engineering
segment-routing
  traffic-engineering
    segment-list name dci1-edge1
      index 1 mpls label 16601
      index 2 mpls label 16501
    policy dci1-edge1-bgp
      color 13401 endpoint 1.0.3.1
      candidate-paths
        preference 30
          explicit segment-list dci1-edge1
```

Verifying Configuration for SRTE Over Default VRF

To display the appropriate details about the SRTE over default VRF configuration, perform one of the following tasks:

Table 4: Verifying SRTE Over Default VRF Configuration

Command	Purpose
show running-config bgp	Displays information about the ingress peer or the SRTE headend.
show running-config bgp	Displays information about the egress peer or the SRTE endpoint.
show running-config srte	Displays information about the SRTE policy for ingress peer.

Additional References

Related Documents

Related Topic	Document Title
BGP	*Cisco Nexus 9000 Series Unicast Routing Configuration Guide*