



## Configure SR-TE Policies

This module provides information about segment routing for traffic engineering (SR-TE) policies, how to configure SR-TE policies, and how to steer traffic into an SR-TE policy.

- [SR-TE Policy Overview, on page 1](#)
- [Usage Guidelines and Limitations, on page 2](#)
- [Instantiation of an SR Policy, on page 2](#)
- [SR-TE Policy Path Types, on page 35](#)
- [Protocols, on page 49](#)
- [Traffic Steering, on page 56](#)
- [Miscellaneous, on page 70](#)

### SR-TE Policy Overview

Segment routing for traffic engineering (SR-TE) uses a “policy” to steer traffic through the network. An SR-TE policy path is expressed as a list of segments that specifies the path, called a segment ID (SID) list. Each segment is an end-to-end path from the source to the destination, and instructs the routers in the network to follow the specified path instead of following the shortest path calculated by the IGP. If a packet is steered into an SR-TE policy, the SID list is pushed on the packet by the head-end. The rest of the network executes the instructions embedded in the SID list.

An SR-TE policy is identified as an ordered list (head-end, color, end-point):

- Head-end – Where the SR-TE policy is instantiated
- Color – A numerical value that distinguishes between two or more policies to the same node pairs (Head-end – End point)
- End-point – The destination of the SR-TE policy

Every SR-TE policy has a color value. Every policy between the same node pairs requires a unique color value.

An SR-TE policy uses one or more candidate paths. A candidate path is a single segment list (SID-list) or a set of weighted SID-lists (for weighted equal cost multi-path [WECCMP]). A candidate path is either dynamic or explicit. See *SR-TE Policy Path Types* section for more information.

# Usage Guidelines and Limitations

Observe the following guidelines and limitations for the platform.

- Before configuring SR-TE policies, use the **distribute link-state** command under IS-IS or OSPF to distribute the link-state database to external services.
- GRE tunnel as primary interface for an SR policy is not supported.
- GRE tunnel as backup interface for an SR policy with TI-LFA protection is not supported.
- Head-end computed inter-domain SR policy with Flex Algo constraint and IGP redistribution is not supported.

## Instantiation of an SR Policy

An SR policy is instantiated, or implemented, at the head-end router.

The following sections provide details on the SR policy instantiation methods:

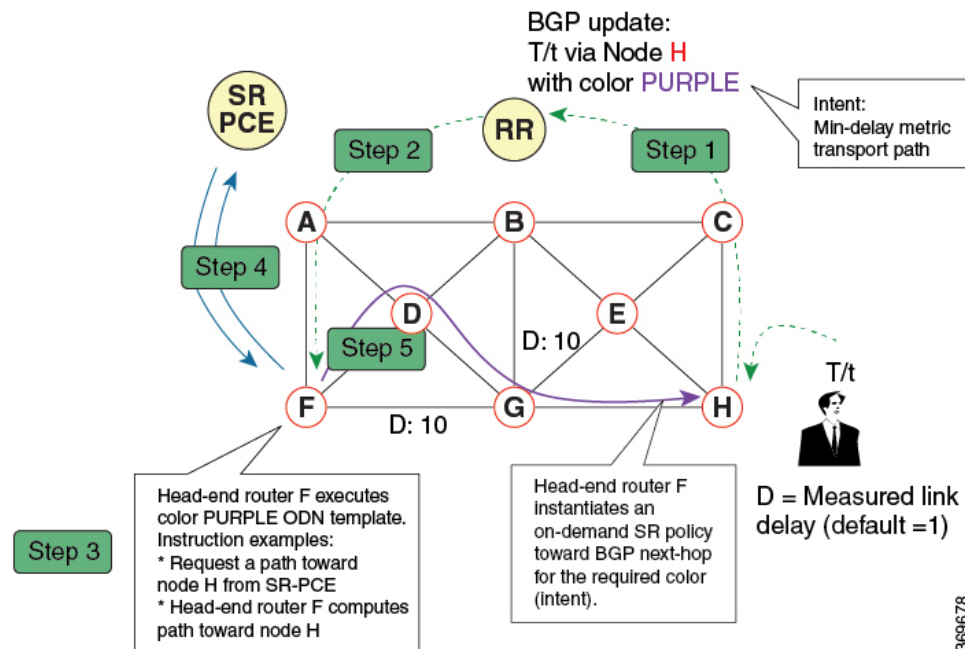
- [On-Demand SR Policy – SR On-Demand Next-Hop](#) , on page 2
- [Manually Provisioned SR Policy](#), on page 34
- [PCE-Initiated SR Policy](#), on page 35

## On-Demand SR Policy – SR On-Demand Next-Hop

Segment Routing On-Demand Next Hop (SR-ODN) allows a service head-end router to automatically instantiate an SR policy to a BGP next-hop when required (on-demand). Its key benefits include:

- **SLA-aware BGP service** – Provides per-destination steering behaviors where a prefix, a set of prefixes, or all prefixes from a service can be associated with a desired underlay SLA. The functionality applies equally to single-domain and multi-domain networks.
- **Simplicity** – No prior SR Policy configuration needs to be configured and maintained. Instead, operator simply configures a small set of common intent-based optimization templates throughout the network.
- **Scalability** – Device resources at the head-end router are used only when required, based on service or SLA connectivity needs.

The following example shows how SR-ODN works:



1. An egress PE (node H) advertises a BGP route for prefix T/t. This advertisement includes an SLA intent encoded with a BGP color extended community. In this example, the operator assigns color purple (example value = 100) to prefixes that should traverse the network over the delay-optimized path.
2. The route reflector receives the advertised route and advertises it to other PE nodes.
3. Ingress PEs in the network (such as node F) are pre-configured with an ODN template for color purple that provides the node with the steps to follow in case a route with the intended color appears, for example:
  - Contact SR-PCE and request computation for a path toward node H that does not share any nodes with another LSP in the same disjointness group.
  - At the head-end router, compute a path towards node H that minimizes cumulative delay.
4. In this example, the head-end router contacts the SR-PCE and requests computation for a path toward node H that minimizes cumulative delay.
5. After SR-PCE provides the compute path, an intent-driven SR policy is instantiated at the head-end router. Other prefixes with the same intent (color) and destined to the same egress PE can share the same on-demand SR policy. When the last prefix associated with a given [intent, egress PE] pair is withdrawn, the on-demand SR policy is deleted, and resources are freed from the head-end router.

An on-demand SR policy is created dynamically for BGP global or VPN (service) routes. The following services are supported with SR-ODN:

- IPv4 BGP global routes
- IPv6 BGP global routes (6PE)
- VPNv4
- VPNv6 (6vPE)
- EVPN-VPWS (single-homing)

## SR-ODN/Automated Steering Support at ASBR for L3VPN Inter-AS Option B and L3VPN Inline Route Reflector

This feature augments support for SR-ODN and automated steering (AS) for the following scenarios:

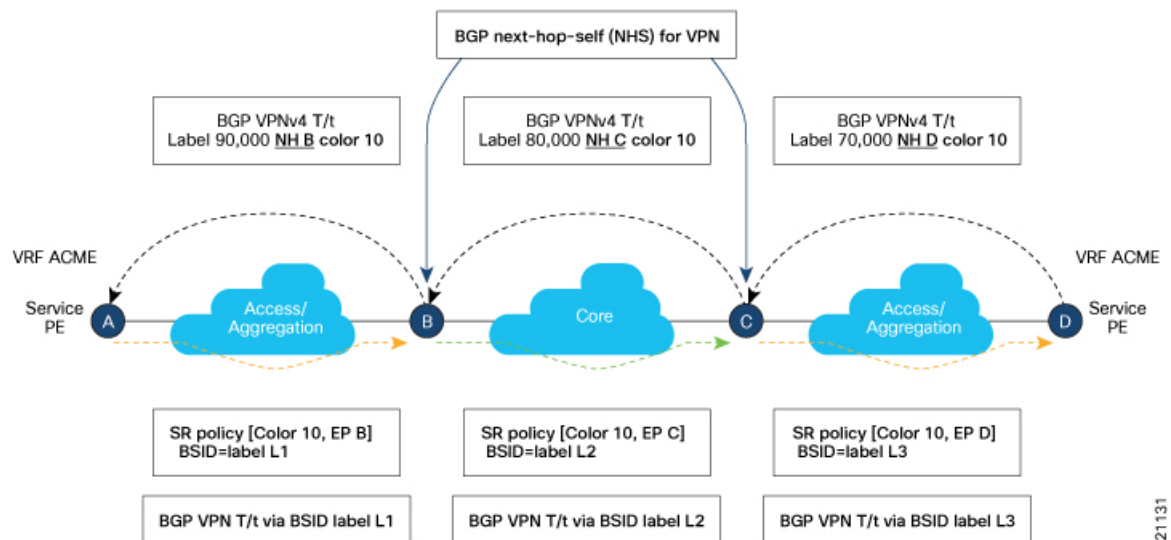
- At ASBR nodes for L3VPN Inter-AS Option B
- At ABR nodes acting as L3VPN Inline Route Reflectors

With this feature, an ABR/ASBR node can trigger an on-demand SR policy used to steer traffic to remote colored destinations.



**Note** This feature is not supported when the Inter-AS Option B Per Next-Hop Label Allocation feature is enabled using the **label mode per-nexthop-received-label** command under the VPNv4 unicast address-family.

The below topology shows a network with different regions under a single BGP AS and with L3VPN services end-to-end. Nodes B and C are ABRs configured with BGP next-hop-self (NHS) for L3VPN. These ABRs program label cross connects for L3VPN destinations.



BGP advertises prefixes with SLA intent by attaching a color extended community. The objective is to steer traffic towards colored VPN prefixes with SR policies in *each* region. As shown in the figure, this feature allows ABR node B to steer traffic for remote BGP prefixes with color 10 over an SR policy with {color 10, end-point C}.

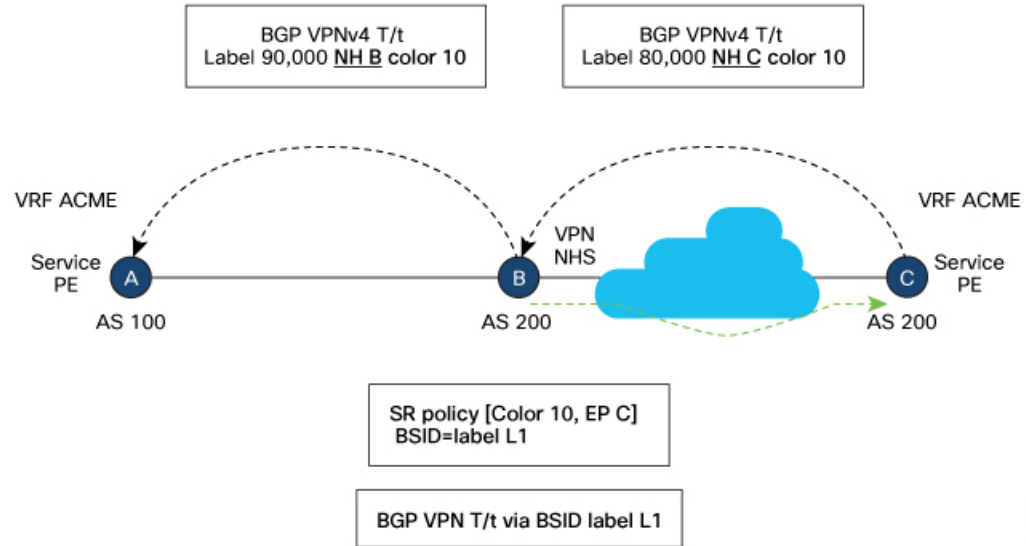
Similar behaviors apply at ASBR nodes for L3VPN Inter-AS Option B scenarios.

### Configuring SR-ODN/AS at L3VPN ABR/ASBR

See the [SR-ODN Configuration Steps, on page 7](#) section for information about configuring the On-Demand Color Template.

### Example – SR-ODN/AS at ASBR for L3VPN Inter-AS Option B

The following example depicts an L3VPN Inter-AS Option B scenario with node B acting as ASBR. Node B steers traffic for remote BGP prefix T/t with color 10 over an on-demand SR policy with a path to node C minimizing the TE metric.

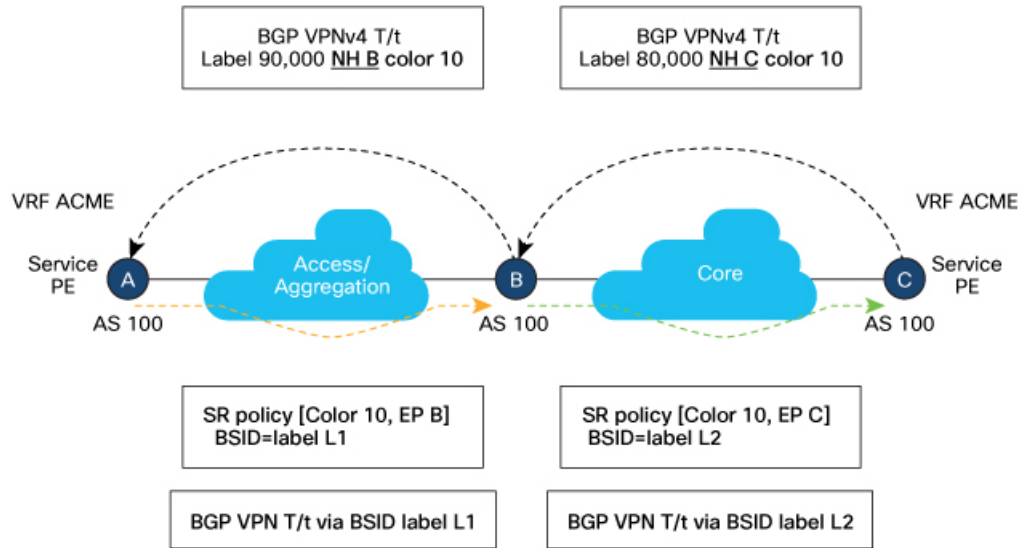


### Configuration on ASBR Node B

```
segment-routing
 traffic-eng
  on-demand color 10
  dynamic
   metric
    type te
  !
  !
  !
  !
router bgp 200
 address-family vpnv4 unicast
  retain route-target all
  !
 neighbor <neighbor_A>
  remote-as 100
  address-family vpnv4 unicast
  send-extended-community-ebgp
  route-policy pass-all in
  route-policy pass-all out
  !
 neighbor <neighbor_C>
  remote-as 200
  address-family vpnv4 unicast
  update-source Loopback0
  next-hop-self
```

### Example - SR-ODN/AS at L3VPN ABR (BGP inline Route Reflector)

The following example depicts a scenario with node B acting as L3VPN BGP inline Route Reflector. Node B steers traffic for remote BGP prefix T/t with color 10 over an on-demand SR policy with a path to node C minimizing the delay metric.



521133

### Configuration on ASBR Node B

```

segment-routing
traffic-eng
  on-demand color 10
  dynamic
  metric
  type latency
  !
  !
  !
  !
router bgp 100
  ibgp policy out enforce-modifications
  address-family vpnv4 unicast
  !
  neighbor <neighbor_C>
  remote-as 100
  address-family vpnv4 unicast
  update-source Loopback0
  route-reflector-client
  next-hop-self
  !
  !
  neighbor <neighbor_A>
  remote-as 100
  address-family vpnv4 unicast
  update-source Loopback0
  route-reflector-client
  next-hop-self

```

## SR-ODN Configuration Steps

To configure SR-ODN, complete the following configurations:

1. Define the SR-ODN template on the SR-TE head-end router.  
(Optional) If using Segment Routing Path Computation Element (SR-PCE) for path computation:
  - a. Configure SR-PCE. For detailed SR-PCE configuration information, see [Configure SR-PCE](#).
  - b. Configure the head-end router as Path Computation Element Protocol (PCEP) Path Computation Client (PCC). For detailed PCEP PCC configuration information, see [Configure the Head-End Router as PCEP PCC](#).
2. Define BGP color extended communities. Refer to the "Implementing BGP" chapter in the [Routing Configuration Guide for Cisco ASR 9000 Series Routers](#).
3. Define routing policies (using routing policy language [RPL]) to set BGP color extended communities. Refer to the "Implementing Routing Policy" chapter in the [Routing Configuration Guide for Cisco ASR 9000 Series Routers](#).

The following RPL attach-points for setting/matching BGP color extended communities are supported:



**Note** The following table shows the supported RPL match operations; however, routing policies are required primarily to set BGP color extended community. Matching based on BGP color extended communities is performed automatically by ODN's on-demand color template.

Attach Point	Set	Match
VRF export	X	X
VRF import	–	X
Neighbor-in	X	X
Neighbor-out	X	X
Inter-AFI export	–	X
Inter-AFI import	–	X
Default-originate	X	–

4. Apply routing policies to a service. Refer to the "Implementing Routing Policy" chapter in the [Routing Configuration Guide for Cisco ASR 9000 Series Routers](#).

### Configure On-Demand Color Template

- Use the **on-demand color** *color* command to create an ODN template for the specified color value. The head-end router automatically follows the actions defined in the template upon arrival of BGP global or VPN routes with a BGP color extended community that matches the color value specified in the template.

The *color* range is from 1 to 4294967295.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 10
```




---

**Note** Matching based on BGP color extended communities is performed automatically via ODN's on-demand color template. RPL routing policies are not required.

---

- Use the **on-demand color *color dynamic*** command to associate the template with on-demand SR policies with a locally computed dynamic path (by SR-TE head-end router utilizing its TE topology database) or centrally (by SR-PCE). The head-end router will first attempt to install the locally computed path; otherwise, it will use the path computed by the SR-PCE.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 10 dynamic
```

- Use the **on-demand color *color dynamic pcep*** command to indicate that only the path computed by SR-PCE should be associated with the on-demand SR policy. With this configuration, local path computation is not attempted; instead the head-end router will only instantiate the path computed by the SR-PCE.

```
Router(config-sr-te)# on-demand color 10 dynamic pcep
```

### Configure Dynamic Path Optimization Objectives

- Use the **metric type {igp | te | latency}** command to configure the metric for use in path computation.

```
Router(config-sr-te-color-dyn)# metric type te
```

- Use the **metric margin {absolute *value* | relative *percent*}** command to configure the On-Demand dynamic path metric margin. The range for *value* and *percent* is from 0 to 2147483647.

```
Router(config-sr-te-color-dyn)# metric margin absolute 5
```

### Configure Dynamic Path Constraints

- Use the **disjoint-path group-id *group-id* type {link | node | srlg | srlg-node} [sub-id *sub-id*]** command to configure the disjoint-path constraints. The *group-id* and *sub-id* range is from 1 to 65535.

```
Router(config-sr-te-color-dyn)# disjoint-path group-id 775 type link
```

- Use the **affinity {include-any | include-all | exclude-any} {name *WORD*}** command to configure the affinity constraints.

```
Router(config-sr-te-color-dyn)# affinity exclude-any name CROSS
```

- Use the **maximum-sid-depth *value*** command to customize the maximum SID depth (MSD) constraints advertised by the router.

The default MSD *value* is equal to the maximum MSD supported by the platform (10).

```
Router(config-sr-te-color)# maximum-sid-depth 5
```



See [Customize MSD Value at PCC, on page 51](#) for information about SR-TE label imposition capabilities.

- Use the **sid-algorithm** *algorithm-number* command to configure the SR Flexible Algorithm constraints. The *algorithm-number* range is from 128 to 255.

```
Router(config-sr-te-color-dyn)# sid-algorithm 128
```

## Configuring SR-ODN: Examples

### Configuring SR-ODN: Layer-3 Services Examples

The following examples show end-to-end configurations used in implementing SR-ODN on the head-end router.

#### Configuring ODN Color Templates: Example

Configure ODN color templates on routers acting as SR-TE head-end nodes. The following example shows various ODN color templates:

- color 10: minimization objective = te-metric
- color 20: minimization objective = igp-metric
- color 21: minimization objective = igp-metric; constraints = affinity
- color 22: minimization objective = te-metric; path computation at SR-PCE; constraints = affinity
- color 30: minimization objective = delay-metric
- color 128: constraints = flex-algo

```
segment-routing
traffic-eng
on-demand color 10
dynamic
metric
type te
!
!
!
on-demand color 20
dynamic
metric
type igp
!
!
!
on-demand color 21
dynamic
metric
type igp
!
affinity exclude-any
name CROSS
!
!
!
on-demand color 22
dynamic
pcep
```

```

!
metric
  type te
!
affinity exclude-any
  name CROSS
!
!
!
on-demand color 30
  dynamic
  metric
    type latency
!
!
on-demand color 128
  dynamic
  sid-algorithm 128
!
!
end

```

### Configuring BGP Color Extended Community Set: Example

The following example shows how to configure BGP color extended communities that are later applied to BGP service routes via route-policies.




---

**Note** In most common scenarios, egress PE routers that advertise BGP service routes apply (set) BGP color extended communities. However, color can also be set at the ingress PE router.

---

```

extcommunity-set opaque color10-te
  10
end-set
!
extcommunity-set opaque color20-igp
  20
end-set
!
extcommunity-set opaque color21-igp-excl-cross
  21
end-set
!
extcommunity-set opaque color30-delay
  30
end-set
!
extcommunity-set opaque color128-fa128
  128
end-set
!

```

### Configuring RPL to Set BGP Color (Layer-3 Services): Examples

The following example shows various representative RPL definitions that set BGP color community.

The first 4 RPL examples include the set color action only. The last RPL example performs the set color action for selected destinations based on a prefix-set.

```

route-policy SET_COLOR_LOW_LATENCY_TE
  set extcommunity color color10-te
  pass
end-policy
!
route-policy SET_COLOR_HI_BW
  set extcommunity color color20-igp
  pass
end-policy
!
route-policy SET_COLOR_LOW_LATENCY
  set extcommunity color color30-delay
  pass
end-policy
!
route-policy SET_COLOR_FA_128
  set extcommunity color color128-fa128
  pass
end-policy
!

prefix-set sample-set
  88.1.0.0/24
end-set
!
route-policy SET_COLOR_GLOBAL
  if destination in sample-set then
    set extcommunity color color10-te
  else
    pass
  endif
end-policy

```

### Applying RPL to BGP Services (Layer-3 Services): Example

The following example shows various RPLs that set BGP color community being applied to BGP Layer-3 VPN services (VPNv4/VPNv6) and BGP global.

- The L3VPN examples show the RPL applied at the VRF export attach-point.
- The BGP global example shows the RPL applied at the BGP neighbor-out attach-point.

```

vrf vrf_cust1
  address-family ipv4 unicast
    export route-policy SET_COLOR_LOW_LATENCY_TE
  !
  address-family ipv6 unicast
    export route-policy SET_COLOR_LOW_LATENCY_TE
  !
!
vrf vrf_cust2
  address-family ipv4 unicast
    export route-policy SET_COLOR_HI_BW
  !
  address-family ipv6 unicast
    export route-policy SET_COLOR_HI_BW
  !
!
vrf vrf_cust3
  address-family ipv4 unicast
    export route-policy SET_COLOR_LOW_LATENCY
  !
  address-family ipv6 unicast

```

```

    export route-policy SET_COLOR_LOW_LATENCY
    !
    !
vrf vrf_cust4
  address-family ipv4 unicast
    export route-policy SET_COLOR_FA_128
    !
  address-family ipv6 unicast
    export route-policy SET_COLOR_FA_128
    !
  !
router bgp 100
  neighbor-group BR-TO-RR
  address-family ipv4 unicast
    route-policy SET_COLOR_GLOBAL out
  !
  !
end

```

### Verifying BGP VRF Information

Use the **show bgp vrf** command to display BGP prefix information for VRF instances. The following output shows the BGP VRF table including a prefix (88.1.1.0/24) with color 10 advertised by router 10.1.1.8.

```

RP/0/RP0/CPU0:R4# show bgp vrf vrf_cust1

BGP VRF vrf_cust1, state: Active
BGP Route Distinguisher: 10.1.1.4:101
VRF ID: 0x60000007
BGP router identifier 10.1.1.4, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000007  RD version: 282
BGP main routing table version 287
BGP NSR Initial initsync version 31 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.4:101 (default for vrf vrf_cust1)
*> 44.1.1.0/24      40.4.101.11          0 400 {1} i
*>i55.1.1.0/24      10.1.1.5              100 0 500 {1} i
*>i88.1.1.0/24      10.1.1.8 C:10         100 0 800 {1} i
*>i99.1.1.0/24      10.1.1.9              100 0 800 {1} i

Processed 4 prefixes, 4 paths

```

The following output displays the details for prefix 88.1.1.0/24. Note the presence of BGP extended color community 10, and that the prefix is associated with an SR policy with color 10 and BSID value of 24036.

```

RP/0/RP0/CPU0:R4# show bgp vrf vrf_cust1 88.1.1.0/24

BGP routing table entry for 88.1.1.0/24, Route Distinguisher: 10.1.1.4:101
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          282      282
Last Modified: May 20 09:23:34.112 for 00:06:03
Paths: (1 available, best #1)
  Advertised to CE peers (in unique update groups):

```

```

40.4.101.11
Path #1: Received by speaker 0
Advertised to CE peers (in unique update groups):
40.4.101.11
800 {1}
10.1.1.8 C:10 (bsid:24036) (metric 20) from 10.1.1.55 (10.1.1.8)
  Received Label 24012
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported
  Received Path ID 0, Local Path ID 1, version 273
  Extended community: Color:10 RT:100:1
  Originator: 10.1.1.8, Cluster list: 10.1.1.55
  SR policy color 10, up, registered, bsid 24036, if-handle 0x08000024

Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 10.1.1.8:101

```

### Verifying Forwarding (CEF) Table

Use the `show cef vrf` command to display the contents of the CEF table for the VRF instance. Note that prefix 88.1.1.0/24 points to the BSID label corresponding to an SR policy. Other non-colored prefixes, such as 55.1.1.0/24, point to BGP next-hop.

```
RP/0/RP0/CPU0:R4# show cef vrf vrf_cust1
```

Prefix	Next Hop	Interface
0.0.0.0/0	drop	default handler
0.0.0.0/32	broadcast	
40.4.101.0/24	attached	TenGigE0/0/0/0.101
40.4.101.0/32	broadcast	TenGigE0/0/0/0.101
40.4.101.4/32	receive	TenGigE0/0/0/0.101
40.4.101.11/32	40.4.101.11/32	TenGigE0/0/0/0.101
40.4.101.255/32	broadcast	TenGigE0/0/0/0.101
44.1.1.0/24	40.4.101.11/32	<recursive>
55.1.1.0/24	10.1.1.5/32	<recursive>
<b>88.1.1.0/24</b>	<b>24036 (via-label)</b>	<b>&lt;recursive&gt;</b>
99.1.1.0/24	10.1.1.9/32	<recursive>
224.0.0.0/4	0.0.0.0/32	
224.0.0.0/24	receive	
255.255.255.255/32	broadcast	

The following output displays CEF details for prefix 88.1.1.0/24. Note that the prefix is associated with an SR policy with BSID value of 24036.

```
RP/0/RP0/CPU0:R4# show cef vrf vrf_cust1 88.1.1.0/24
```

```

88.1.1.0/24, version 51, internal 0x5000001 0x0 (ptr 0x98c60ddc) [1], 0x0 (0x0), 0x208
(0x98425268)
Updated May 20 09:23:34.216
Prefix Len 24, traffic index 0, precedence n/a, priority 3
via local-label 24036, 5 dependencies, recursive [flags 0x6000]
path-idx 0 NHID 0x0 [0x97091ec0 0x0]
recursion-via-label
next hop VRF - 'default', table - 0xe0000000
next hop via 24036/0/21
next hop srte_c_10_ep labels imposed {ImplNull 24012}

```

### Verifying SR Policy

Use the `show segment-routing traffic-eng policy` command to display SR policy information.

The following outputs show the details of an on-demand SR policy that was triggered by prefixes with color 10 advertised by node 10.1.1.8.

```
RP/0/RP0/CPU0:R4# show segment-routing traffic-eng policy color 10 tabular
```

Color	Endpoint	Admin State	Oper State	Binding SID
10	10.1.1.8	up	up	24036

The following outputs show the details of the on-demand SR policy for BSID 24036.



**Note** There are 2 candidate paths associated with this SR policy: the path that is computed by the head-end router (with preference 200), and the path that is computed by the SR-PCE (with preference 100). The candidate path with the highest preference is the active candidate path (highlighted below) and is installed in forwarding.

```
RP/0/RP0/CPU0:R4# show segment-routing traffic-eng policy binding-sid 24036
```

```
SR-TE policy database
```

```
-----
Color: 10, End-point: 10.1.1.8
Name: srte_c_10_ep_10.1.1.8
Status:
  Admin: up Operational: up for 4d14h (since Jul  3 20:28:57.840)
Candidate-paths:
  Preference: 200 (BGP ODN) (active)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_10_ep_10.1.1.8_discr_200
      PLSP-ID: 12
    Dynamic (valid)
      Metric Type: TE, Path Accumulated Metric: 30
        16009 [Prefix-SID, 10.1.1.9]
        16008 [Prefix-SID, 10.1.1.8]
  Preference: 100 (BGP ODN)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_10_ep_10.1.1.8_discr_100
      PLSP-ID: 11
    Dynamic (pce 10.1.1.57) (valid)
      Metric Type: TE, Path Accumulated Metric: 30
        16009 [Prefix-SID, 10.1.1.9]
        16008 [Prefix-SID, 10.1.1.8]
Attributes:
  Binding SID: 24036
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes
```

### Verifying SR Policy Forwarding

Use the **show segment-routing traffic-eng forwarding policy** command to display the SR policy forwarding information.

The following outputs show the forwarding details for an on-demand SR policy that was triggered by prefixes with color 10 advertised by node 10.1.1.8.

```

RP/0/RP0/CPU0:R4# show segment-routing traffic-eng forwarding policy binding-sid 24036
tabular

Color Endpoint          Segment      Outgoing Outgoing   Next Hop   Bytes   Pure
                        List         Label     Interface Next Hop   Switched Backup
-----
10    10.1.1.8           dynamic     16009    Gi0/0/0/4  10.4.5.5   0       0
                        16001    Gi0/0/0/5  11.4.8.8   0       Yes

RP/0/RP0/CPU0:R4# show segment-routing traffic-eng forwarding policy binding-sid 24036
detail
Mon Jul  8 11:56:46.887 PST

SR-TE Policy Forwarding database
-----

Color: 10, End-point: 10.1.1.8
  Name: srte_c_10_ep_10.1.1.8
  Binding SID: 24036
  Segment Lists:
    SL[0]:
      Name: dynamic
      Paths:
        Path[0]:
          Outgoing Label: 16009
          Outgoing Interface: GigabitEthernet0/0/0/4
          Next Hop: 10.4.5.5
          Switched Packets/Bytes: 0/0
          FRR Pure Backup: No
          Label Stack (Top -> Bottom): { 16009, 16008 }
          Path-id: 1 (Protected), Backup-path-id: 2, Weight: 64
        Path[1]:
          Outgoing Label: 16001
          Outgoing Interface: GigabitEthernet0/0/0/5
          Next Hop: 11.4.8.8
          Switched Packets/Bytes: 0/0
          FRR Pure Backup: Yes
          Label Stack (Top -> Bottom): { 16001, 16009, 16008 }
          Path-id: 2 (Pure-Backup), Weight: 64
  Policy Packets/Bytes Switched: 0/0
  Local label: 80013

```

## Configuring SR-ODN for EVPN-VPWS: Use Case

### Configuration: SR-PCE

For cases when PCC nodes support, or signal, PCEP association-group object to indicate the pair of LSPs in a disjoint set, there is no extra configuration required at the SR-PCE to trigger disjoint-path computation.



**Note** SR-PCE also supports disjoint-path computation for cases when PCC nodes do not support PCEP association-group object. See [Configure the Disjoint Policy \(Optional\)](#) for more information.

### Configuration: Site 1 Node A

This section depicts relevant configuration of Node A at Site 1. It includes service configuration, BGP color extended community, and RPL. It also includes the corresponding ODN template required to achieve the disjointness SLA.

Nodes in Site 1 are configured to set color 10000 on originating EVPN routes, while matching color 11000 on incoming EVPN routes from routers located at Site 2.

Since both nodes in Site 1 request path computation from SR-PCE using the same disjoint-path group-id (775), the PCE will attempt to compute disjointness for the pair of LSPs originating from Site 1 toward Site 2.

```

/* EVPN-VPWS configuration */

interface GigabitEthernet0/0/0/3.2500 l2transport
 encapsulation dot1q 2500
 rewrite ingress tag pop 1 symmetric
!
l2vpn
xconnect group evpn_vpws_group
 p2p evpn_vpws_100
  interface GigabitEthernet0/0/0/3.2500
   neighbor evpn evi 100 target 21 source 11
  !
!
!
!

/* BGP color community and RPL configuration */

extcommunity-set opaque color-10000
 10000
end-set
!
route-policy SET_COLOR_EVPN_VPWS
 if evpn-route-type is 1 and rd in (ios-regex '.*..*..*..*(100)') then
  set extcommunity color color-10000
 endif
 pass
end-policy
!
router bgp 65000
 neighbor 10.1.1.253
  address-family l2vpn evpn
  route-policy SET_COLOR_EVPN_VPWS out
!
!
!

/* ODN template configuration */

segment-routing
traffic-eng
 on-demand color 11000
 dynamic
  pcep
  !
  metric
  type igp
  !
  disjoint-path group-id 775 type link
  !
!
!
!

```



**Configuration: Site 1 Node B**

This section depicts relevant configuration of Node B at Site 1.

```

/* EVPN-VPWS configuration */

interface TenGigE0/3/0/0/8.2500 l2transport
 encapsulation dot1q 2500
 rewrite ingress tag pop 1 symmetric
 !
l2vpn
 xconnect group evpn_vpws_group
  p2p evpn_vpws_101
   interface TenGigE0/3/0/0/8.2500
    neighbor evpn evi 101 target 22 source 12
   !
  !
 !
 !

/* BGP color community and RPL configuration */

extcommunity-set opaque color-10000
 10000
end-set
!
route-policy SET_COLOR_EVPN_VPWS
 if evpn-route-type is 1 and rd in (ios-regex '.*..*..*..*(101)') then
  set extcommunity color color-10000
 endif
 pass
end-policy
!
router bgp 65000
 neighbor 10.1.1.253
  address-family l2vpn evpn
   route-policy SET_COLOR_EVPN_VPWS out
  !
 !
 !

/* ODN template configuration */

segment-routing
 traffic-eng
  on-demand color 11000
  dynamic
   pcep
   !
   metric
    type igp
   !
   disjoint-path group-id 775 type link
  !
 !
 !
 !

```

**Configuration: Site 2 Node C**

This section depicts relevant configuration of Node C at Site 2. It includes service configuration, BGP color extended community, and RPL. It also includes the corresponding ODN template required to achieve the disjointness SLA.

Nodes in Site 2 are configured to set color 11000 on originating EVPN routes, while matching color 10000 on incoming EVPN routes from routers located at Site 1.

Since both nodes on Site 2 request path computation from SR-PCE using the same disjoint-path group-id (776), the PCE will attempt to compute disjointness for the pair of LSPs originating from Site 2 toward Site 1.

```

/* EVPN-VPWS configuration */

interface GigabitEthernet0/0/0/3.2500 l2transport
 encapsulation dot1q 2500
 rewrite ingress tag pop 1 symmetric
 !
l2vpn
xconnect group evpn_vpws_group
 p2p evpn_vpws_100
  interface GigabitEthernet0/0/0/3.2500
   neighbor evpn evi 100 target 11 source 21
  !
 !
 !
 !

/* BGP color community and RPL configuration */

extcommunity-set opaque color-11000
 11000
end-set
!
route-policy SET_COLOR_EVPN_VPWS
 if evpn-route-type is 1 and rd in (ios-regex '.*..*..*..*:(100)') then
  set extcommunity color color-11000
 endif
 pass
end-policy
!
router bgp 65000
 neighbor 10.1.1.253
  address-family l2vpn evpn
  route-policy SET_COLOR_EVPN_VPWS out
 !
 !
 !

/* ODN template configuration */

segment-routing
traffic-eng
 on-demand color 10000
 dynamic
  pcep
  !
  metric
  type igp
  !
 disjoint-path group-id 776 type link
 !
 !
 !
 !

```

**Configuration: Site 2 Node D**

This section depicts relevant configuration of Node D at Site 2.

```

/* EVPN-VPWS configuration */

interface GigabitEthernet0/0/0/1.2500 l2transport
 encapsulation dot1q 2500
 rewrite ingress tag pop 1 symmetric
 !
l2vpn
 xconnect group evpn_vpws_group
  p2p evpn_vpws_101
   interface GigabitEthernet0/0/0/1.2500
    neighbor evpn evi 101 target 12 source 22
   !
  !
 !
 !

/* BGP color community and RPL configuration */

extcommunity-set opaque color-11000
 11000
end-set
!
route-policy SET_COLOR_EVPN_VPWS
 if evpn-route-type is 1 and rd in (ios-regex '.*..*..*..*(101)') then
  set extcommunity color color-11000
 endif
 pass
end-policy
!
router bgp 65000
 neighbor 10.1.1.253
  address-family l2vpn evpn
  route-policy SET_COLOR_EVPN_VPWS out
 !
 !
 !

/* ODN template configuration */

segment-routing
 traffic-eng
  on-demand color 10000
  dynamic
  pcep
  !
  metric
  type igp
  !
  disjoint-path group-id 776 type link
 !
 !
 !
 !

```

**Verification: SR-PCE**

Use the **show pce ipv4 peer** command to display the SR-PCE's PCEP peers and session status. SR-PCE performs path computation for the 4 nodes depicted in the use-case.

```
RP/0/0/CPU0:SR-PCE# show pce ipv4 peer
Mon Jul 15 19:41:43.622 UTC

PCE's peer database:
-----
Peer address: 10.1.1.2
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Peer address: 10.1.1.4
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Peer address: 10.1.1.5
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation

Peer address: 10.1.1.6
  State: Up
  Capabilities: Stateful, Segment-Routing, Update, Instantiation
```

Use the **show pce association group-id** command to display information for the pair of LSPs assigned to a given association group-id value.

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. In particular, disjoint LSPs from site 1 to site 2 are identified by association group-id 775. The output includes high-level information for LSPs associated to this group-id:

- At Node A (10.1.1.5): LSP symbolic name = `bgp_c_11000_ep_10.1.1.2_discr_100`
- At Node B (10.1.1.6): LSP symbolic name = `bgp_c_11000_ep_10.1.1.4_discr_100`

In this case, the SR-PCE was able to achieve the desired disjointness level; therefore the Status is shown as "Satisfied".

```
RP/0/0/CPU0:SR-PCE# show pce association group-id 775
Thu Jul 11 03:52:20.770 UTC

PCE's association database:
-----
Association: Type Link-Disjoint, Group 775, Not Strict
Associated LSPs:
  LSP[0]:
    PCC 10.1.1.6, tunnel name bgp_c_11000_ep_10.1.1.4_discr_100, PLSP ID 18, tunnel ID 17,
    LSP ID 3, Configured on PCC
  LSP[1]:
    PCC 10.1.1.5, tunnel name bgp_c_11000_ep_10.1.1.2_discr_100, PLSP ID 18, tunnel ID 18,
    LSP ID 3, Configured on PCC
Status: Satisfied
```

Use the **show pce lsp** command to display detailed information of an LSP present in the PCE's LSP database. This output shows details for the LSP at Node A (10.1.1.5) that is used to carry traffic of EVPN VPWS EVI 100 towards node C (10.1.1.2).

```
RP/0/0/CPU0:SR-PCE# show pce lsp pcc ipv4 10.1.1.5 name bgp_c_11000_ep_10.1.1.2_discr_100
Thu Jul 11 03:58:45.903 UTC

PCE's tunnel database:
-----
PCC 10.1.1.5:

Tunnel Name: bgp_c_11000_ep_10.1.1.2_discr_100
Color: 11000
```

```

Interface Name: srte_c_11000_ep_10.1.1.2
LSPs:
LSP[0]:
  source 10.1.1.5, destination 10.1.1.2, tunnel ID 18, LSP ID 3
  State: Admin up, Operation up
  Setup type: Segment Routing
  Binding SID: 80037
  Maximum SID Depth: 10
  Absolute Metric Margin: 0
  Relative Metric Margin: 0%
  Preference: 100
  Bandwidth: signaled 0 kbps, applied 0 kbps
  PCEP information:
    PLSP-ID 0x12, flags: D:1 S:0 R:0 A:1 O:1 C:0
  LSP Role: Exclude LSP
  State-sync PCE: None
  PCC: 10.1.1.5
  LSP is subdelegated to: None
  Reported path:
    Metric type: IGP, Accumulated Metric 40
    SID[0]: Adj, Label 80003, Address: local 11.5.8.5 remote 11.5.8.8
    SID[1]: Node, Label 16007, Address 10.1.1.7
    SID[2]: Node, Label 16002, Address 10.1.1.2
  Computed path: (Local PCE)
  Computed Time: Thu Jul 11 03:49:48 UTC 2019 (00:08:58 ago)
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Adj, Label 80003, Address: local 11.5.8.5 remote 11.5.8.8
  SID[1]: Node, Label 16007, Address 10.1.1.7
  SID[2]: Node, Label 16002, Address 10.1.1.2
  Recorded path:
    None
Disjoint Group Information:
  Type Link-Disjoint, Group 775

```

This output shows details for the LSP at Node B (10.1.1.6) that is used to carry traffic of EVPN VPWS EVI 101 towards node D (10.1.1.4).

```

RP/0/0/CPU0:SR-PCE# show pce lsp pcc ipv4 10.1.1.6 name bgp_c_11000_ep_10.1.1.4_discr_100
Thu Jul 11 03:58:56.812 UTC

```

PCE's tunnel database:

```

-----
PCC 10.1.1.6:

```

```

Tunnel Name: bgp_c_11000_ep_10.1.1.4_discr_100

```

```

Color: 11000

```

```

Interface Name: srte_c_11000_ep_10.1.1.4

```

```

LSPs:
LSP[0]:
  source 10.1.1.6, destination 10.1.1.4, tunnel ID 17, LSP ID 3
  State: Admin up, Operation up
  Setup type: Segment Routing
  Binding SID: 80061
  Maximum SID Depth: 10
  Absolute Metric Margin: 0
  Relative Metric Margin: 0%
  Preference: 100
  Bandwidth: signaled 0 kbps, applied 0 kbps
  PCEP information:
    PLSP-ID 0x12, flags: D:1 S:0 R:0 A:1 O:1 C:0
  LSP Role: Disjoint LSP
  State-sync PCE: None
  PCC: 10.1.1.6
  LSP is subdelegated to: None

```

```

Reported path:
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16001, Address 10.1.1.1
  SID[1]: Node, Label 16004, Address 10.1.1.4
Computed path: (Local PCE)
  Computed Time: Thu Jul 11 03:49:48 UTC 2019 (00:09:08 ago)
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16001, Address 10.1.1.1
  SID[1]: Node, Label 16004, Address 10.1.1.4
Recorded path:
  None
Disjoint Group Information:
  Type Link-Disjoint, Group 775

```

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. In particular, disjoint LSPs from site 2 to site 1 are identified by association group-id 776. The output includes high-level information for LSPs associated to this group-id:

- At Node C (10.1.1.2): LSP symbolic name = `bgp_c_10000_ep_10.1.1.5_discr_100`
- At Node D (10.1.1.4): LSP symbolic name = `bgp_c_10000_ep_10.1.1.6_discr_100`

In this case, the SR-PCE was able to achieve the desired disjointness level; therefore, the Status is shown as "Satisfied".

```

RP/0/0/CPU0:SR-PCE# show pce association group-id 776
Thu Jul 11 03:52:24.370 UTC

```

```

PCE's association database:
-----

```

```

Association: Type Link-Disjoint, Group 776, Not Strict

```

```

Associated LSPs:
LSP[0]:
  PCC 10.1.1.4, tunnel name bgp_c_10000_ep_10.1.1.6_discr_100, PLSP ID 16, tunnel ID 14,
  LSP ID 1, Configured on PCC
LSP[1]:
  PCC 10.1.1.2, tunnel name bgp_c_10000_ep_10.1.1.5_discr_100, PLSP ID 6, tunnel ID 21,
  LSP ID 3, Configured on PCC
Status: Satisfied

```

Use the `show pce lsp` command to display detailed information of an LSP present in the PCE's LSP database. This output shows details for the LSP at Node C (10.1.1.2) that is used to carry traffic of EVPN VPWS EVI 100 towards node A (10.1.1.5).

```

RP/0/0/CPU0:SR-PCE# show pce lsp pcc ipv4 10.1.1.2 name bgp_c_10000_ep_10.1.1.5_discr_100
Thu Jul 11 03:55:21.706 UTC

```

```

PCE's tunnel database:
-----

```

```

PCC 10.1.1.2:

```

```

Tunnel Name: bgp_c_10000_ep_10.1.1.5_discr_100

```

```

Color: 10000

```

```

Interface Name: srte_c_10000_ep_10.1.1.5

```

```

LSPs:

```

```

LSP[0]:
  source 10.1.1.2, destination 10.1.1.5, tunnel ID 21, LSP ID 3
  State: Admin up, Operation up
  Setup type: Segment Routing
  Binding SID: 80052
  Maximum SID Depth: 10
  Absolute Metric Margin: 0
  Relative Metric Margin: 0%

```

```

Preference: 100
Bandwidth: signaled 0 kbps, applied 0 kbps
PCEP information:
  PLSP-ID 0x6, flags: D:1 S:0 R:0 A:1 O:1 C:0
LSP Role: Exclude LSP
State-sync PCE: None
PCC: 10.1.1.2
LSP is subdelegated to: None
Reported path:
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16007, Address 10.1.1.7
  SID[1]: Node, Label 16008, Address 10.1.1.8
  SID[2]: Adj, Label 80005, Address: local 11.5.8.8 remote 11.5.8.5
Computed path: (Local PCE)
  Computed Time: Thu Jul 11 03:50:03 UTC 2019 (00:05:18 ago)
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16007, Address 10.1.1.7
  SID[1]: Node, Label 16008, Address 10.1.1.8
  SID[2]: Adj, Label 80005, Address: local 11.5.8.8 remote 11.5.8.5
Recorded path:
  None
Disjoint Group Information:
  Type Link-Disjoint, Group 776

```

This output shows details for the LSP at Node D (10.1.1.4) used to carry traffic of EVPN VPWS EVI 101 towards node B (10.1.1.6).

```

RP/0/0/CPU0:SR-PCE# show pce lsp pcc ipv4 10.1.1.4 name bgp_c_10000_ep_10.1.1.6_discr_100
Thu Jul 11 03:55:23.296 UTC

```

PCE's tunnel database:

-----  
PCC 10.1.1.4:

Tunnel Name: bgp\_c\_10000\_ep\_10.1.1.6\_discr\_100

**Color: 10000**

**Interface Name: srte\_c\_10000\_ep\_10.1.1.6**

LSPs:

LSP[0]:

```

  source 10.1.1.4, destination 10.1.1.6, tunnel ID 14, LSP ID 1
  State: Admin up, Operation up
  Setup type: Segment Routing
  Binding SID: 80047
  Maximum SID Depth: 10
  Absolute Metric Margin: 0
  Relative Metric Margin: 0%
  Preference: 100
  Bandwidth: signaled 0 kbps, applied 0 kbps
PCEP information:
  PLSP-ID 0x10, flags: D:1 S:0 R:0 A:1 O:1 C:0
LSP Role: Disjoint LSP
State-sync PCE: None
PCC: 10.1.1.4
LSP is subdelegated to: None
Reported path:
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16001, Address 10.1.1.1
  SID[1]: Node, Label 16006, Address 10.1.1.6
Computed path: (Local PCE)
  Computed Time: Thu Jul 11 03:50:03 UTC 2019 (00:05:20 ago)
  Metric type: IGP, Accumulated Metric 40
  SID[0]: Node, Label 16001, Address 10.1.1.1
  SID[1]: Node, Label 16006, Address 10.1.1.6
Recorded path:

```

```

None
Disjoint Group Information:
  Type Link-Disjoint, Group 776

```

### Verification: Site 1 Node A

This section depicts verification steps at Node A.

Use the **show bgp l2vpn evpn** command to display BGP prefix information for EVPN-VPWS EVI 100 (rd 10.1.1.5:100). The output includes an EVPN route-type 1 route with color 11000 originated at Node C (10.1.1.2).

```

RP/0/RSP0/CPU0:Node-A# show bgp l2vpn evpn rd 10.1.1.5:100
Wed Jul 10 18:57:57.704 PST
BGP router identifier 10.1.1.5, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 360
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.5:100 (default for vrf VPWS:100)
*> [1][0000.0000.0000.0000.0000][11]/120
                               0.0.0.0                0 i
*>i[1][0000.0000.0000.0000][21]/120
                               10.1.1.2 C:11000          100      0 i

```

The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 11000, and that the prefix is associated with an SR policy with color 11000 and BSID value of 80044.

```

RP/0/RSP0/CPU0:Node-A# show bgp l2vpn evpn rd 10.1.1.5:100
[1][0000.0000.0000.0000.0000][21]/120
Wed Jul 10 18:57:58.107 PST
BGP routing table entry for [1][0000.0000.0000.0000.0000][21]/120, Route Distinguisher:
10.1.1.5:100
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          360        360
Last Modified: Jul 10 18:36:18.369 for 00:21:40
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
Local
  10.1.1.2 C:11000 (bsid:80044) (metric 40) from 10.1.1.253 (10.1.1.2)
  Received Label 80056
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported, rib-install
  Received Path ID 0, Local Path ID 1, version 358
  Extended community: Color:11000 RT:65000:100
  Originator: 10.1.1.2, Cluster list: 10.1.1.253
  SR policy color 11000, up, registered, bsid 80044, if-handle 0x00001b20

Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 10.1.1.2:100

```



Use the **show l2vpn xconnect** command to display the state associated with EVPN-VPWS EVI 100 service.

```
RP/0/RSP0/CPU0:Node-A# show l2vpn xconnect group evpn_vpws_group
Wed Jul 10 18:58:02.333 PST
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

XConnect
Group      Name      ST      Segment 1      ST      Segment 2      ST
-----
evpn_vpws_group
          evpn_vpws_100
                        UP      Gi0/0/0/3.2500      UP      EVPN 100,21,10.1.1.2      UP
```

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 11000 and end-point 10.1.1.2 (node C).

```
RP/0/RSP0/CPU0:Node-A# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_100
detail
Wed Jul 10 18:58:02.755 PST

Group evpn_vpws_group, XC evpn_vpws_100, state is up; Interworking none
AC: GigabitEthernet0/0/0/3.2500, state is up
Type VLAN; Num Ranges: 1
Rewrite Tags: []
VLAN ranges: [2500, 2500]
MTU 1500; XC ID 0x120000c; interworking none
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0
  drops: illegal VLAN 0, illegal length 0
EVPN: neighbor 10.1.1.2, PW ID: evi 100, ac-id 21, state is up ( established )
XC ID 0xa0000007
Encapsulation MPLS
Source address 10.1.1.5
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active : SR TE srte_c_11000_ep_10.1.1.2, On-Demand, fallback enabled
Tunnel : Up
Load Balance Hashing: src-dst-mac

      EVPN      Local      Remote
-----
Label      80040      80056
MTU        1500      1500
Control word enabled
AC ID      11        21
EVPN type  Ethernet  Ethernet

-----
Create time: 10/07/2019 18:31:30 (1d17h ago)
Last time status changed: 10/07/2019 19:42:00 (1d16h ago)
Last time PW went down: 10/07/2019 19:40:55 (1d16h ago)
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0
```

Use the **show segment-routing traffic-eng policy** command with **tabular** option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80044 that was triggered by EVPN RT1 prefix with color 11000 advertised by node C (10.1.1.2).

```
RP/0/RSP0/CPU0:Node-A# show segment-routing traffic-eng policy color 11000 tabular
Wed Jul 10 18:58:00.732 PST
```

Color	Endpoint	Admin State	Oper State	Binding SID
11000	10.1.1.2	up	up	80044

The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (10.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 1 to site 2, LSP at Node A (srte\_c\_11000\_ep\_10.1.1.2) is link-disjoint from LSP at Node B (srte\_c\_11000\_ep\_10.1.1.4).

```
RP/0/RSP0/CPU0:Node-A# show segment-routing traffic-eng policy color 11000
Wed Jul 10 19:15:47.217 PST
```

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

```
Color: 11000, End-point: 10.1.1.2
Name: srte_c_11000_ep_10.1.1.2
Status:
  Admin: up Operational: up for 00:39:31 (since Jul 10 18:36:00.471)
Candidate-paths:
  Preference: 200 (BGP ODN) (shutdown)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_11000_ep_10.1.1.2_discr_200
      PLSP-ID: 19
    Dynamic (invalid)
  Preference: 100 (BGP ODN) (active)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_11000_ep_10.1.1.2_discr_100
      PLSP-ID: 18
    Dynamic (pce 10.1.1.207) (valid)
      Metric Type: IGP, Path Accumulated Metric: 40
      80003 [Adjacency-SID, 11.5.8.5 - 11.5.8.8]
      16007 [Prefix-SID, 10.1.1.7]
      16002 [Prefix-SID, 10.1.1.2]
Attributes:
  Binding SID: 80044
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes
```

### Verification: Site 1 Node B

This section depicts verification steps at Node B.

Use the **show bgp l2vpn evpn** command to display BGP prefix information for EVPN-VPWS EVI 101 (rd 10.1.1.6:101). The output includes an EVPN route-type 1 route with color 11000 originated at Node D (10.1.1.4).

```
RP/0/RSP0/CPU0:Node-B# show bgp l2vpn evpn rd 10.1.1.6:101
Wed Jul 10 19:08:54.964 PST
BGP router identifier 10.1.1.6, local AS number 65000
```

```

BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 322
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.6:101 (default for vrf VPWS:101)
*> [1][0000.0000.0000.0000.0000][12]/120
                0.0.0.0                                0 i
*>i[1][0000.0000.0000.0000.0000][22]/120
                10.1.1.4 C:11000                        100      0 i

Processed 2 prefixes, 2 paths
    
```

The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 11000, and that the prefix is associated with an SR policy with color 11000 and BSID value of 80061.

```

RP/0/RSP0/CPU0:Node-B# show bgp l2vpn evpn rd 10.1.1.6:101
[1][0000.0000.0000.0000.0000][22]/120
Wed Jul 10 19:08:55.039 PST
BGP routing table entry for [1][0000.0000.0000.0000.0000][22]/120, Route Distinguisher:
10.1.1.6:101
Versions:
  Process          bRIB/RIB    SendTblVer
  Speaker          322        322
Last Modified: Jul 10 18:42:10.408 for 00:26:44
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
Local
  10.1.1.4 C:11000 (bsid:80061) (metric 40) from 10.1.1.253 (10.1.1.4)
  Received Label 80045
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported, rib-install
  Received Path ID 0, Local Path ID 1, version 319
  Extended community: Color:11000 RT:65000:101
  Originator: 10.1.1.4, Cluster list: 10.1.1.253
  SR policy color 11000, up, registered, bsid 80061, if-handle 0x00000560

Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 10.1.1.4:101
    
```

Use the **show l2vpn xconnect** command to display the state associated with EVPN-VPWS EVI 101 service.

```

RP/0/RSP0/CPU0:Node-B# show l2vpn xconnect group evpn_vpws_group
Wed Jul 10 19:08:56.388 PST
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

XConnect
Group      Name      ST      Segment 1
-----
          Description      ST      Segment 2
          Description      ST
evpn_vpws_group
    
```

```

evpn_vpws_101
UP Te0/3/0/0/8.2500 UP EVPN 101,22,10.1.1.4 UP
-----

```

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 11000 and end-point 10.1.1.4 (node D).

```

RP/0/RSP0/CPU0:Node-B# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_101
Wed Jul 10 19:08:56.511 PST

```

```

Group evpn_vpws_group, XC evpn_vpws_101, state is up; Interworking none
AC: TenGigE0/3/0/0/8.2500, state is up
Type VLAN; Num Ranges: 1
Rewrite Tags: []
VLAN ranges: [2500, 2500]
MTU 1500; XC ID 0x2a0000e; interworking none
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0
  drops: illegal VLAN 0, illegal length 0
EVPN: neighbor 10.1.1.4, PW ID: evi 101, ac-id 22, state is up ( established )
XC ID 0xa0000009
Encapsulation MPLS
Source address 10.1.1.6
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active : SR TE srte_c_11000_ep_10.1.1.4, On-Demand, fallback enabled
Tunnel : Up
Load Balance Hashing: src-dst-mac

```

EVPN	Local	Remote
Label	80060	80045
MTU	1500	1500
Control word	enabled	enabled
AC ID	12	22
EVPN type	Ethernet	Ethernet

```

-----
Create time: 10/07/2019 18:32:49 (00:36:06 ago)
Last time status changed: 10/07/2019 18:42:07 (00:26:49 ago)
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0

```

Use the **show segment-routing traffic-eng policy** command with **tabular** option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80061 that was triggered by EVPN RT1 prefix with color 11000 advertised by node D (10.1.1.4).

```

RP/0/RSP0/CPU0:Node-B# show segment-routing traffic-eng policy color 11000 tabular
Wed Jul 10 19:08:56.146 PST

```

Color	Endpoint	Admin State	Oper State	Binding SID
11000	10.1.1.4	up	up	80061

The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (10.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 1 to site 2, LSP at Node B (srte\_c\_11000\_ep\_10.1.1.4) is link-disjoint from LSP at Node A (srte\_c\_11000\_ep\_10.1.1.2).

```
RP/0/RSP0/CPU0:Node-B# show segment-routing traffic-eng policy color 11000
Wed Jul 10 19:08:56.207 PST
```

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

```
Color: 11000, End-point: 10.1.1.4
Name: srte_c_11000_ep_10.1.1.4
Status:
  Admin: up   Operational: up for 00:26:47 (since Jul 10 18:40:05.868)
Candidate-paths:
  Preference: 200 (BGP ODN) (shutdown)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_11000_ep_10.1.1.4_discr_200
      PLSP-ID: 19
      Dynamic (invalid)
Preference: 100 (BGP ODN) (active)
Requested BSID: dynamic
  PCC info:
    Symbolic name: bgp_c_11000_ep_10.1.1.4_discr_100
    PLSP-ID: 18
Dynamic (pce 10.1.1.207) (valid)
Metric Type: IGP, Path Accumulated Metric: 40
16001 [Prefix-SID, 10.1.1.1]
16004 [Prefix-SID, 10.1.1.4]
Attributes:
Binding SID: 80061
Forward Class: 0
Steering BGP disabled: no
IPv6 caps enable: yes
```

### Verification: Site 2 Node C

This section depicts verification steps at Node C.

Use the **show bgp l2vpn evpn** command to display BGP prefix information for EVPN-VPWS EVI 100 (rd 10.1.1.2:100). The output includes an EVPN route-type 1 route with color 10000 originated at Node A (10.1.1.5).

```
RP/0/RSP0/CPU0:Node-C# show bgp l2vpn evpn rd 10.1.1.2:100
BGP router identifier 10.1.1.2, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 21
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.2:100 (default for vrf VPWS:100)
*>i[1][0000.0000.0000.0000.0000][11]/120
                10.1.1.5 C:10000                100      0 i
*> [1][0000.0000.0000.0000.0000][21]/120
```

0.0.0.0

0 i

The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 10000, and that the prefix is associated with an SR policy with color 10000 and BSID value of 80058.

```
RP/0/RSP0/CPU0:Node-C# show bgp l2vpn evpn rd 10.1.1.2:100
[1][0000.0000.0000.0000.0000][11]/120
BGP routing table entry for [1][0000.0000.0000.0000.0000][11]/120, Route Distinguisher:
10.1.1.2:100
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          20        20
Last Modified: Jul 10 18:36:20.503 for 00:45:21
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
Local
  10.1.1.5 C:10000 (bsid:80058) (metric 40) from 10.1.1.253 (10.1.1.5)
  Received Label 80040
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported, rib-install
  Received Path ID 0, Local Path ID 1, version 18
  Extended community: Color:10000 RT:65000:100
  Originator: 10.1.1.5, Cluster list: 10.1.1.253
  SR policy color 10000, up, registered, bsid 80058, if-handle 0x000006a0

  Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 10.1.1.5:100
```

Use the **show l2vpn xconnect** command to display the state associated with EVPN-VPWS EVI 100 service.

```
RP/0/RSP0/CPU0:Node-C# show l2vpn xconnect group evpn_vpws_group
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
       SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
```

XConnect Group	Name	ST	Segment 1 Description	ST	Segment 2 Description	ST
evpn_vpws_group	evpn_vpws_100	UP	Gi0/0/0/3.2500	UP	EVPN 100,11,10.1.1.5	UP

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 10000 and end-point 10.1.1.5 (node A).

```
RP/0/RSP0/CPU0:Node-C# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_100
```

```
Group evpn_vpws_group, XC evpn_vpws_100, state is up; Interworking none
AC: GigabitEthernet0/0/0/3.2500, state is up
  Type VLAN; Num Ranges: 1
  Rewrite Tags: []
  VLAN ranges: [2500, 2500]
  MTU 1500; XC ID 0x1200008; interworking none
  Statistics:
    packets: received 0, sent 0
    bytes: received 0, sent 0
    drops: illegal VLAN 0, illegal length 0
  EVPN: neighbor 10.1.1.5, PW ID: evi 100, ac-id 11, state is up ( established )
  XC ID 0xa0000003
```

```

Encapsulation MPLS
Source address 10.1.1.2
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active : SR TE srte_c_10000_ep_10.1.1.5, On-Demand, fallback enabled
Tunnel : Up
Load Balance Hashing: src-dst-mac

```

EVPN	Local	Remote
Label	80056	80040
MTU	1500	1500
Control word	enabled	enabled
AC ID	21	11
EVPN type	Ethernet	Ethernet

```

-----
Create time: 10/07/2019 18:36:16 (1d19h ago)
Last time status changed: 10/07/2019 19:41:59 (1d18h ago)
Last time PW went down: 10/07/2019 19:40:54 (1d18h ago)
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0

```

Use the **show segment-routing traffic-eng policy** command with **tabular** option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80058 that was triggered by EVPN RT1 prefix with color 10000 advertised by node A (10.1.1.5).

```
RP/0/RSP0/CPU0:Node-C# show segment-routing traffic-eng policy color 10000 tabular
```

Color	Endpoint	Admin State	Oper State	Binding SID
10000	10.1.1.5	up	up	80058

The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (10.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 2 to site 1, LSP at Node C (srte\_c\_10000\_ep\_10.1.1.5) is link-disjoint from LSP at Node D (srte\_c\_10000\_ep\_10.1.1.6).

```
RP/0/RSP0/CPU0:Node-C# show segment-routing traffic-eng policy color 10000
```

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

```

Color: 10000, End-point: 10.1.1.5
Name: srte_c_10000_ep_10.1.1.5
Status:
  Admin: up Operational: up for 00:12:35 (since Jul 10 19:49:21.890)
Candidate-paths:
  Preference: 200 (BGP ODN) (shutdown)
  Requested BSID: dynamic
  PCC info:
    Symbolic name: bgp_c_10000_ep_10.1.1.5_discr_200
    PLSP-ID: 7
    Dynamic (invalid)
Preference: 100 (BGP ODN) (active)
  Requested BSID: dynamic

```

```

PCC info:
  Symbolic name: bgp_c_10000_ep_10.1.1.5_discr_100
  PLSP-ID: 6
  Dynamic (pce 10.1.1.207) (valid)
    Metric Type: IGP, Path Accumulated Metric: 40
    16007 [Prefix-SID, 10.1.1.7]
    16008 [Prefix-SID, 10.1.1.8]
    80005 [Adjacency-SID, 11.5.8.8 - 11.5.8.5]
Attributes:
  Binding SID: 80058
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

```

### Verification: Site 2 Node D

This section depicts verification steps at Node D.

Use the **show bgp l2vpn evpn** command to display BGP prefix information for EVPN-VPWS EVI 101 (rd 10.1.1.4:101). The output includes an EVPN route-type 1 route with color 10000 originated at Node B (10.1.1.6).

```

RP/0/RSP0/CPU0:Node-D# show bgp l2vpn evpn rd 10.1.1.4:101
BGP router identifier 10.1.1.4, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 570
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.4:101 (default for vrf VPWS:101)
*>i[1][0000.0000.0000.0000.0000][12]/120
                10.1.1.6 C:10000                100      0 i
*> [1][0000.0000.0000.0000.0000][22]/120
                0.0.0.0                                0 i

Processed 2 prefixes, 2 paths

```

The following output displays the details for the incoming EVPN RT1. Note the presence of BGP extended color community 10000, and that the prefix is associated with an SR policy with color 10000 and BSID value of 80047.

```

RP/0/RSP0/CPU0:Node-D# show bgp l2vpn evpn rd 10.1.1.4:101
[1][0000.0000.0000.0000.0000][12]/120
BGP routing table entry for [1][0000.0000.0000.0000.0000][12]/120, Route Distinguisher:
10.1.1.4:101
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          569      569
Last Modified: Jul 10 18:42:12.455 for 00:45:38
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local

```



```

10.1.1.6 C:10000 (bsid:80047) (metric 40) from 10.1.1.253 (10.1.1.6)
  Received Label 80060
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
imported, rib-install
  Received Path ID 0, Local Path ID 1, version 568
Extended community: Color:10000 RT:65000:101
  Originator: 10.1.1.6, Cluster list: 10.1.1.253
SR policy color 10000, up, registered, bsid 80047, if-handle 0x00001720

Source AFI: L2VPN EVPN, Source VRF: default, Source Route Distinguisher: 10.1.1.6:101
    
```

Use the **show l2vpn xconnect** command to display the state associated with EVPN-VPWS EVI 101 service.

```

RP/0/RSP0/CPU0:Node-D# show l2vpn xconnect group evpn_vpws_group
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
       SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

XConnect
Group      Name          ST      Segment 1
-----
          Description          ST      Segment 2
          Description          ST
-----
evpn_vpws_group
          evpn_vpws_101
          UP      Gi0/0/0/1.2500          UP      EVPN 101,12,10.1.1.6          UP
-----
    
```

The following output shows the details for the service. Note that the service is associated with the on-demand SR policy with color 10000 and end-point 10.1.1.6 (node B).

```

RP/0/RSP0/CPU0:Node-D# show l2vpn xconnect group evpn_vpws_group xc-name evpn_vpws_101

Group evpn_vpws_group, XC evpn_vpws_101, state is up; Interworking none
AC: GigabitEthernet0/0/0/1.2500, state is up
Type VLAN; Num Ranges: 1
Rewrite Tags: []
VLAN ranges: [2500, 2500]
MTU 1500; XC ID 0x120000c; interworking none
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0
  drops: illegal VLAN 0, illegal length 0
EVPN: neighbor 10.1.1.6, PW ID: evi 101, ac-id 12, state is up ( established )
XC ID 0xa000000d
Encapsulation MPLS
Source address 10.1.1.4
Encap type Ethernet, control word enabled
Sequencing not set
Preferred path Active : SR TE srte_c_10000_ep_10.1.1.6, On-Demand, fallback enabled
Tunnel : Up
Load Balance Hashing: src-dst-mac

EVPN          Local          Remote
-----
Label          80045          80060
MTU            1500          1500
Control word   enabled        enabled
AC ID          22            12
EVPN type     Ethernet      Ethernet

-----
Create time: 10/07/2019 18:42:07 (00:45:49 ago)
Last time status changed: 10/07/2019 18:42:09 (00:45:47 ago)
Statistics:
    
```

```

packets: received 0, sent 0
bytes: received 0, sent 0

```

Use the **show segment-routing traffic-eng policy** command with **tabular** option to display SR policy summary information.

The following output shows the on-demand SR policy with BSID 80047 that was triggered by EVPN RT1 prefix with color 10000 advertised by node B (10.1.1.6).

```
RP/0/RSP0/CPU0:Node-D# show segment-routing traffic-eng policy color 10000 tabular
```

Color	Endpoint	Admin State	Oper State	Binding SID
10000	10.1.1.6	up	up	80047

The following output shows the details for the on-demand SR policy. Note that the SR policy's active candidate path (preference 100) is computed by SR-PCE (10.1.1.207).

Based on the goals of this use case, SR-PCE computes link-disjoint paths for the SR policies associated with a pair of ELINE services between site 1 and site 2. Specifically, from site 2 to site 1, LSP at Node D (srte\_c\_10000\_ep\_10.1.1.6) is link-disjoint from LSP at Node C (srte\_c\_10000\_ep\_10.1.1.5).

```
RP/0/RSP0/CPU0:Node-D# show segment-routing traffic-eng policy color 10000
```

```
SR-TE policy database
```

```

-----
Color: 10000, End-point: 10.1.1.6
Name: srte_c_10000_ep_10.1.1.6
Status:
Admin: up Operational: up for 01:23:04 (since Jul 10 18:42:07.350)
Candidate-paths:
Preference: 200 (BGP ODN) (shutdown)
Requested BSID: dynamic
PCC info:
Symbolic name: bgp_c_10000_ep_10.1.1.6_discr_200
PLSP-ID: 17
Dynamic (invalid)
Preference: 100 (BGP ODN) (active)
Requested BSID: dynamic
PCC info:
Symbolic name: bgp_c_10000_ep_10.1.1.6_discr_100
PLSP-ID: 16
Dynamic (pce 10.1.1.207) (valid)
Metric Type: IGP, Path Accumulated Metric: 40
16001 [Prefix-SID, 10.1.1.1]
16006 [Prefix-SID, 10.1.1.6]
Attributes:
Binding SID: 80047
Forward Class: 0
Steering BGP disabled: no
IPv6 caps enable: yes

```

## Manually Provisioned SR Policy

Manually provisioned SR policies are configured on the head-end router. These policies can use dynamic paths or explicit paths. See the [SR-TE Policy Path Types, on page 35](#) section for information on manually provisioning an SR policy using dynamic or explicit paths.

## PCE-Initiated SR Policy

An SR-TE policy can be configured on the path computation element (PCE) to reduce link congestion or to minimize the number of network touch points.

The PCE collects network information, such as traffic demand and link utilization. When the PCE determines that a link is congested, it identifies one or more flows that are causing the congestion. The PCE finds a suitable path and deploys an SR-TE policy to divert those flows, without moving the congestion to another part of the network. When there is no more link congestion, the policy is removed.

To minimize the number of network touch points, an application, such as a Network Services Orchestrator (NSO), can request the PCE to create an SR-TE policy. PCE deploys the SR-TE policy using PCC-PCE communication protocol (PCEP).

For more information, see the [PCE-Initiated SR Policies](#) section.

## SR-TE Policy Path Types

A **dynamic** path is based on an optimization objective and a set of constraints. The head-end computes a solution, resulting in a SID-list or a set of SID-lists. When the topology changes, a new path is computed. If the head-end does not have enough information about the topology, the head-end might delegate the computation to a Segment Routing Path Computation Element (SR-PCE). For information on configuring SR-PCE, see *Configure Segment Routing Path Computation Element* chapter.

An **explicit** path is a specified SID-list or set of SID-lists.

An SR-TE policy initiates a single (selected) path in RIB/FIB. This is the preferred valid candidate path.

A candidate path has the following characteristics:

- It has a preference – If two policies have same {color, endpoint} but different preferences, the policy with the highest preference is selected.
- It is associated with a single binding SID (BSID) – A BSID conflict occurs when there are different SR policies with the same BSID. In this case, the policy that is installed first gets the BSID and is selected.
- It is valid if it is usable.

A path is selected when the path is valid and its preference is the best among all candidate paths for that policy.



---

**Note** The protocol of the source is not relevant in the path selection logic.

---

## Dynamic Paths

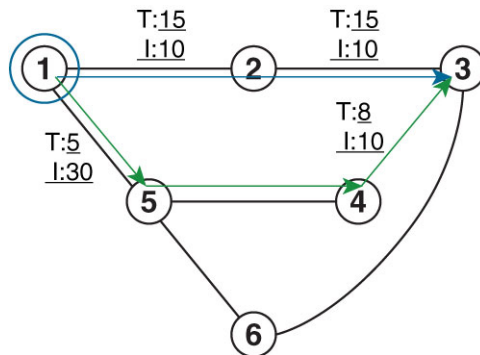
### Optimization Objectives

Optimization objectives allow the head-end router to compute a SID-list that expresses the shortest dynamic path according to the selected metric type:

- IGP metric — Refer to the "Implementing IS-IS" and "Implementing OSPF" chapters in the *Routing Configuration Guide for Cisco ASR 9000 Series Routers*.

- TE metric — See the [Configure Interface TE Metrics, on page 36](#) section for information about configuring TE metrics.

This example shows a dynamic path from head-end router 1 to end-point router 3 that minimizes IGP or TE metric:



Default IGP link metric: I:10  
Default TE link metric T:10

520018

- The blue path uses the minimum IGP metric: Min-Metric (1 → 3, IGP) = SID-list <16003>; cumulative IGP metric: 20
- The green path uses the minimum TE metric: Min-Metric (1 → 3, TE) = SID-list <16005, 16004, 16003>; cumulative TE metric: 23

## Configure Interface TE Metrics

Use the **metric value** command in SR-TE interface submode to configure the TE metric for interfaces. The **value** range is from 0 to 2147483647.

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# interface type interface-path-id
Router(config-sr-te-if)# metric value
```

### Configuring TE Metric: Example

The following configuration example shows how to set the TE metric for various interfaces:

```
segment-routing
traffic-eng
interface TenGigE0/0/0/0
metric 100
!
interface TenGigE0/0/0/1
metric 1000
!
interface TenGigE0/0/2/0
metric 50
!
!
end
```

## Constraints

Constraints allow the head-end router to compute a dynamic path according to the selected metric type:

- **Affinity** — You can apply a color or name to links or interfaces by assigning affinity bit-maps to them. You can then specify an affinity (or relationship) between an SR policy path and link colors. SR-TE computes a path that includes or excludes links that have specific colors, or combinations of colors. See the [Named Interface Link Admin Groups and SR-TE Affinity Maps, on page 37](#) section for information on named interface link admin groups and SR-TE Affinity Maps.
- **Disjoint** — SR-TE computes a path that is disjoint from another path in the same disjoint-group. Disjoint paths do not share network resources. Path disjointness may be required for paths between the same pair of nodes, between different pairs of nodes, or a combination (only same head-end or only same end-point).
- **Flexible Algorithm** — Flexible Algorithm allows for user-defined algorithms where the IGP computes paths based on a user-defined combination of metric type and constraint.

### Named Interface Link Admin Groups and SR-TE Affinity Maps

Named Interface Link Admin Groups and SR-TE Affinity Maps provide a simplified and more flexible means of configuring link attributes and path affinities to compute paths for SR-TE policies.

In the traditional TE scheme, links are configured with attribute-flags that are flooded with TE link-state parameters using Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF).

Named Interface Link Admin Groups and SR-TE Affinity Maps let you assign, or map, up to 32 color names for affinity and attribute-flag attributes instead of 32-bit hexadecimal numbers. After mappings are defined, the attributes can be referred to by the corresponding color name in the CLI. Furthermore, you can define constraints using *include-any*, *include-all*, and *exclude-any* arguments, where each statement can contain up to 10 colors.




---

**Note** You can configure affinity constraints using attribute flags or the Flexible Name Based Policy Constraints scheme; however, when configurations for both schemes exist, only the configuration pertaining to the new scheme is applied.

---

### Configure Named Interface Link Admin Groups and SR-TE Affinity Maps

Use the **affinity name** *NAME* command in SR-TE interface submode to assign affinity to interfaces. Configure this on routers with interfaces that have an associated admin group attribute.

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# interface TenGigE0/0/1/2
Router(config-sr-if)# affinity
Router(config-sr-if-affinity)# name RED
```

Use the **affinity-map name** *NAME* **bit-position** *bit-position* command in SR-TE sub-mode to define affinity maps. The *bit-position* range is from 0 to 255.

Configure affinity maps on the following routers:

- Routers with interfaces that have an associated admin group attribute.
- Routers that act as SR-TE head-ends for SR policies that include affinity constraints.

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# affinity-map
Router(config-sr-te-affinity-map)# name RED bit-position 23
```

### Configuring Link Admin Group: Example

The following example shows how to assign affinity to interfaces and to define affinity maps. This configuration is applicable to any router (SR-TE head-end or transit node) with colored interfaces.

```
segment-routing
traffic-eng
interface TenGigE0/0/1/1
  affinity
  name CROSS
  name RED
  !
!
interface TenGigE0/0/1/2
  affinity
  name RED
  !
!
interface TenGigE0/0/2/0
  affinity
  name BLUE
  !
!
affinity-map
name RED bit-position 23
name BLUE bit-position 24
name CROSS bit-position 25
!
end
```

## Configure SR Policy with Dynamic Path

To configure a SR-TE policy with a dynamic path, optimization objectives, and affinity constraints, complete the following configurations:

1. Define the optimization objectives. See the [Optimization Objectives, on page 35](#) section.
2. Define the constraints. See the [Constraints, on page 37](#) section.
3. Create the policy.

### Behaviors and Limitations

#### Examples

The following example shows a configuration of an SR policy at an SR-TE head-end router. The policy has a dynamic path with optimization objectives and affinity constraints computed by the head-end router.

```
segment-routing
traffic-eng
policy foo
color 100 end-point ipv4 10.1.1.2
candidate-paths
```



- Once you enter an MPLS label, you cannot enter an IP address.

When configuring an explicit path using IP addresses of links along the path, the SR-TE process selects either the protected or the unprotected Adj-SID of the link, depending on the order in which the Adj-SIDs were received.

### Configure Local SR-TE Policy Using Explicit Paths

Create a segment list with IP addresses:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list name SIDLIST1
Router(config-sr-te-sl)# index 10 address ipv4 10.1.1.2
Router(config-sr-te-sl)# index 20 address ipv4 10.1.1.3
Router(config-sr-te-sl)# index 30 address ipv4 10.1.1.4
Router(config-sr-te-sl)# exit
```

Create a segment list with MPLS labels:

```
Router(config-sr-te)# segment-list name SIDLIST2
Router(config-sr-te-sl)# index 10 mpls label 16002
Router(config-sr-te-sl)# index 20 mpls label 16003
Router(config-sr-te-sl)# index 30 mpls label 16004
Router(config-sr-te-sl)# exit
```

Create a segment list with invalid MPLS label:

```
Router(config-sr-te)# segment-list name SIDLIST4
Router(config-sr-te-sl)# index 10 mpls label 16009
Router(config-sr-te-sl)# index 20 mpls label 16003
Router(config-sr-te-sl)# index 30 mpls label 16004
Router(config-sr-te-sl)# exit
```

Create a segment list with IP addresses and MPLS labels:

```
Router(config-sr-te)# segment-list name SIDLIST3
Router(config-sr-te-sl)# index 10 address ipv4 10.1.1.2
Router(config-sr-te-sl)# index 20 mpls label 16003
Router(config-sr-te-sl)# index 30 mpls label 16004
Router(config-sr-te-sl)# exit
```

Create the SR-TE policy:

```
Router(config-sr-te)# policy POLICY2
Router(config-sr-te-policy)# color 20 end-point ipv4 10.1.1.4
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST2
Router(config-sr-te-pp-info)# exit
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-policy-path)# preference 200
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST1
Router(config-sr-te-pp-info)# exit
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST4
Router(config-sr-te-pp-info)# exit
Router(config-sr-te-policy-path-pref)# exit
```



## Running Configuration

```
Router# show running-configuration
segment-routing
traffic-eng
segment-list SIDLIST1
  index 10 address ipv4 10.1.1.2
  index 20 address ipv4 10.1.1.3
  index 30 address ipv4 10.1.1.4
!
segment-list SIDLIST2
  index 10 mpls label 16002
  index 20 mpls label 16003
  index 30 mpls label 16004
!
segment-list SIDLIST3
  index 10 address ipv4 10.1.1.2
  index 20 mpls label 16003
  index 30 mpls label 16004
!
segment-list SIDLIST4
  index 10 mpls label 16009
  index 20 mpls label 16003
  index 30 mpls label 16004
!
policy POLICY1
  color 10 end-point ipv4 10.1.1.4
  candidate-paths
  preference 100
    explicit segment-list SIDLIST1
  !
!
!
!
policy POLICY2
  color 20 end-point ipv4 10.1.1.4
  candidate-paths
  preference 100
    explicit segment-list SIDLIST1
  !
!
  preference 200
    explicit segment-list SIDLIST2
  !
    explicit segment-list SIDLIST4
  !
!
!
!
policy POLICY3
  color 30 end-point ipv4 10.1.1.4
  candidate-paths
  preference 100
    explicit segment-list SIDLIST3
  !
!
!
!
!
```

## Verification

Verify the SR-TE policy configuration using:

```
Router# show segment-routing traffic-eng policy name srte_c_20_ep_10.1.1.4
```

```
SR-TE policy database
```

```
-----
Color: 20, End-point: 10.1.1.4
Name: srte_c_20_ep_10.1.1.4
Status:
  Admin: up Operational: up for 00:00:15 (since Jul 14 00:53:10.615)
Candidate-paths:
  Preference: 200 (configuration) (active)
  Name: POLICY2
  Requested BSID: dynamic
  Protection Type: protected-preferred
  Maximum SID Depth: 8
  Explicit: segment-list SIDLIST2 (active)
  Weight: 1, Metric Type: TE
    16002
    16003
    16004
Attributes:
  Binding SID: 51301
  Forward Class: Not Configured
  Steering labeled-services disabled: no
  Steering BGP disabled: no
  IPv6 caps enable: yes
  Invalidation drop enabled: no
```

The following example displays the private output of an SRv6 policy with an explicit path:

```
Router# show segment-routing traffic-eng policy name srte_c_10_ep_fcbb:bb00:2::1 private
```

```
SR-TE policy database
```

```
-----
Color: 10, End-point: fcbb:bb00:2::1 ID: 3
Name: srte_c_10_ep_fcbb:bb00:2::1
Status:
  Admin: up Operational: up for 1w3d (since Jan 14 12:59:41.802)
Candidate-paths:
  Preference: 10 (configuration) (active)
  Originator: ASN 0 node-address <None> discriminator: 10
  Name: srv6_policy
  Requested BSID: dynamic
  Protection Type: protected-preferred
  Maximum SID Depth: 19
  ID: 1
  Source: fcbb:bb00:1::1
  Stale: no
  Checkpoint flags: 0x00000000
  Explicit: segment-list BLUE-SL-1 (valid)
  Weight: 1, Metric Type: TE
  IGP area: 0
    SID[0]:Unknown: fcbb:bb00:10:e0ab::/64 /* Behavior will be optional */
    Format: F3216
    LBL:32 LNL:16 FL:0 AL:0.
    /* NAI not present for explicit path */
    SID[1]:Unknown: fcbb:bb00:e0bc::/48
    Format: F3216
```

```

LBL:32 LNL:16 FL:0 AL:0.
  SID[2]: Unknown: fcbb:bb00:e0cd::/48
  Format: F3216
LBL:32 LNL:16 FL:0 AL:0.
  SID[3]: Unknown: fcbb:bb00:e0d2::/48
  Format: F3216
LBL:32 LNL:16 FL:0 AL:0.
SRv6 Information:
  Locator: loc1
  Binding SID requested: Dynamic
  Binding SID behavior: End.B6.Insert.Red
LSPs:
LSP[0]:
  LSP-ID: 8 policy ID: 3 (active)
  State: Programmed
  Binding SID: fccc:cccl:a1:e006::
  Install timer:
    Running: no
  Cleanup timer:
    Running: no
  Delete timer:
    Running: no
  Revert timer:
    Running: no
  SM chain:
    Init -> BSID RW
    BSID rewrite pending -> Success
  Forwarding flags: 0x00000008
  Candidate path ID: 1
  Flags:
  SLs:
  SL[0]:
    Name: usid_list_1
    Type: Explicit
    NH SRV6 SID: fcbb:bb00:10:e0ab::
    Flags:
    Paths:
    Path[0]:
      Interface version: 1
      Flags:
      Outgoing interface: Gi0/0/0/0
      SID stack: fcbb:bb00:10:e0ab::/64, fcbb:bb00:e0bc::/48,
                fcbb:bb00:e0cd::/48, fcbb:bb00:e0d2::/48
    Path[1]:
      Interface version: 1
      Flags:
      Outgoing interface: Gi0/0/0/1
      SID stack: fcbb:bb00:bb22:e000::/64, fcbb:bb00:10:e0ab::/64,
                fcbb:bb00:e0bc::/48, fcbb:bb00:e0cd::/48,
                fcbb:bb00:e0d2::/48
  Attributes:
    Binding SID: fccc:cccl:a1:e006::
    Forward Class: Not Configured
    Steering labeled-services disabled: no
    Steering BGP disabled: no
    IPv6 caps enable: yes
    Invalidation drop enabled: no
    Max Install Standby Candidate Paths: 0

```

The following example displays the private output of an SRv6 policy forwarding show command:

```

Router# show segment-routing traffic-eng forwarding policy name srte_c_10_ep_fcbb:bb00:2::1
private

```

```

SR-TE Policy Forwarding database
-----
Color: 10, End-point: fcbb:bb00:2::1
  Name: srte_c_10_ep_fcbb:bb00:2::1
  Binding SID: fccc:cccl:a1:e006::
  Active LSP:
    Candidate path:
      Preference: 10 (configuration)
      Originator: ASN 0 node-address <None> discriminator: 10
      Name: srv6_policy
      Segment lists:
        SL[0]:
          Name: usid_list_1
          Switched Packets/Bytes: ??
          Paths:
            Path[0]:
              Outgoing Interfaces: GigabitEthernet0/0/0/0
              Next Hop: fe80::41:dbff:fe02:6513
              FRR Pure Backup: No
              ECMP/LFA Backup: No
              SID stack (Top -> Bottom): { fcbb:bb00:10:e0ab::/64, fcbb:bb00:e0bc::/48,
                                          fcbb:bb00:e0d2::/48,
                                          fcbb:bb00:e0cd::/48}
            Path[1]:
              Outgoing Interfaces: GigabitEthernet0/0/0/1
              Next Hop: fe80::cd:23ff:fe31:1766
              FRR Pure Backup: Yes
              ECMP/LFA Backup: Yes
              SID stack (Top -> Bottom): { fcbb:bb00:bb22:e000::/64, fcbb:bb00:10:e0ab::/64,
                                          fcbb:bb00:e0bc::/48,
                                          fcbb:bb00:e0d2::/48}
          Path-id: 1 (Protected), Backup-path-id: 65, Weight: 0
          Flags: 0x0 (extended 0x0, RIB 0x42)
        Path[1]:
          Outgoing Interfaces: GigabitEthernet0/0/0/1
          Next Hop: fe80::cd:23ff:fe31:1766
          FRR Pure Backup: Yes
          ECMP/LFA Backup: Yes
          SID stack (Top -> Bottom): { fcbb:bb00:bb22:e000::/64, fcbb:bb00:10:e0ab::/64,
                                          fcbb:bb00:e0bc::/48,
                                          fcbb:bb00:e0d2::/48}
          Path-id: 65 (Pure-Backup), Weight: 0
          Flags: 0x0 (extended 0x0, RIB 0x10182)

Policy Packets/Bytes Switched: ??
Interface: None (0x0)

```

## Configuring Explicit Path with Affinity Constraint Validation

To fully configure SR-TE flexible name-based policy constraints, you must complete these high-level tasks in order:

1. Assign Color Names to Numeric Values
2. Associate Affinity-Names with SR-TE Links
3. Associate Affinity Constraints for SR-TE Policies

```

/* Enter the global configuration mode and assign color names to numeric values
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# affinity-map
Router(config-sr-te-affinity-map)# blue bit-position 0
Router(config-sr-te-affinity-map)# green bit-position 1
Router(config-sr-te-affinity-map)# red bit-position 2
Router(config-sr-te-affinity-map)# exit

```

```

/* Associate affinity-names with SR-TE links
Router(config-sr-te)# interface Gi0/0/0/0
Router(config-sr-te-if)# affinity
Router(config-sr-te-if-affinity)# blue
Router(config-sr-te-if-affinity)# exit
Router(config-sr-te-if)# exit
Router(config-sr-te)# interface Gi0/0/0/1
Router(config-sr-te-if)# affinity
Router(config-sr-te-if-affinity)# blue
Router(config-sr-te-if-affinity)# green
Router(config-sr-te-if-affinity)# exit
Router(config-sr-te-if)# exit
Router(config-sr-te)#

/* Associate affinity constraints for SR-TE policies
Router(config-sr-te)# segment-list name SIDLIST1
Router(config-sr-te-sl)# index 10 address ipv4 10.1.1.2
Router(config-sr-te-sl)# index 20 address ipv4 2.2.2.23
Router(config-sr-te-sl)# index 30 address ipv4 10.1.1.4
Router(config-sr-te-sl)# exit
Router(config-sr-te)# segment-list name SIDLIST2
Router(config-sr-te-sl)# index 10 address ipv4 10.1.1.2
Router(config-sr-te-sl)# index 30 address ipv4 10.1.1.4
Router(config-sr-te-sl)# exit
Router(config-sr-te)# segment-list name SIDLIST3
Router(config-sr-te-sl)# index 10 address ipv4 10.1.1.5
Router(config-sr-te-sl)# index 30 address ipv4 10.1.1.4
Router(config-sr-te-sl)# exit

Router(config-sr-te)# policy POLICY1
Router(config-sr-te-policy)# color 20 end-point ipv4 10.1.1.4
Router(config-sr-te-policy)# binding-sid mpls 1000
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 200
Router(config-sr-te-policy-path-pref)# constraints affinity exclude-any red
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST1
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST2
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-policy-path-pref)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST3

```

## Running Configuration

```

Router# show running-configuration
segment-routing
traffic-eng

interface GigabitEthernet0/0/0/0
  affinity
  blue
  !
!
interface GigabitEthernet0/0/0/1
  affinity
  blue
  green

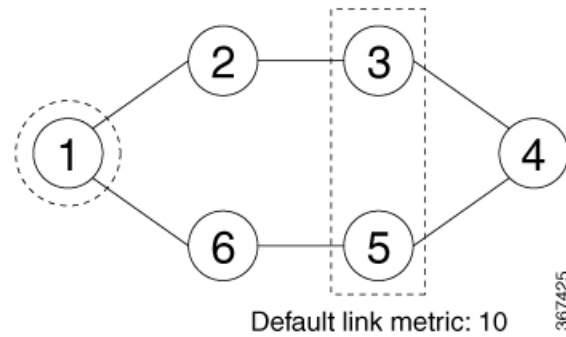
```



The shortest path with the lowest IGP metric is then verified against the affinity constraints. If multiple nodes have the same shortest-path metric, all their paths are validated against the affinity constraints. A path that is not the shortest path is not validated against the affinity constraints.

### Affinity Support for Anycast SIDs: Examples

In the following examples, nodes 3 and 5 advertise the same Anycast prefix (10.1.1.8) and assign the same prefix SID (16100).

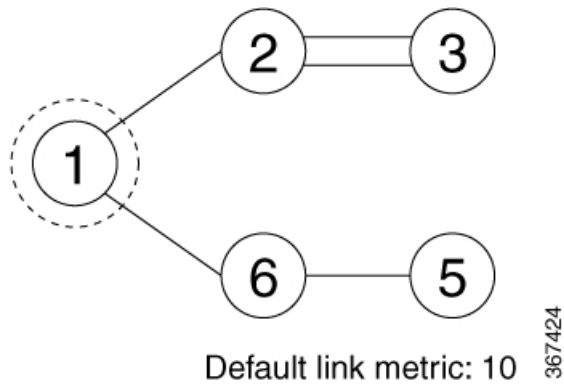


Node 1 uses the following SR-TE policy:

```
segment-routing
traffic-eng
policy POLICY1
  color 20 end-point ipv4 10.1.1.4
  binding-sid mpls 1000
  candidate-paths
  preference 100
  explicit segment-list SIDLIST1
  constraints
  affinity
  exclude-any
  red
segment-list name SIDLIST1
  index 10 address ipv4 100.100.100.100
  index 20 address ipv4 4.4.4.4
```

### Affinity Constraint Validation With ECMP Anycast SID: Example

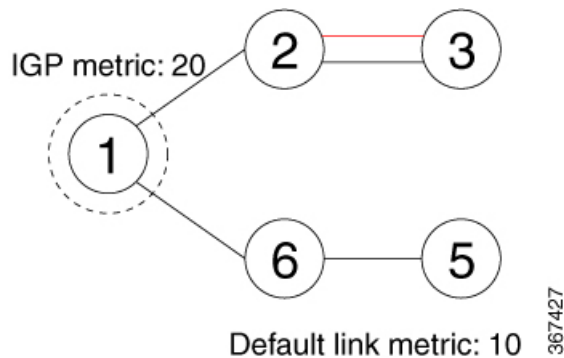
In this example, the shortest path to both node 3 and node 5 has an equal accumulative IGP metric of 20. Both paths are validated against affinity constraints.



```
Name: POLICY1 (Color: 2, End-point: 198.51.100.6)
Status:
  Admin: up Operational: up for 00:03:52 (since Jan 24 01:52:14.215)
Candidate-paths:
  Preference 100:
  Constraints:
  Affinity:
    exclude-any: red
  Explicit: segment-list SIDLIST1 (active)
  Weight: 0, Metric Type: IGP
    16100 [Prefix-SID, 10.1.1.8]
    16004 [Prefix-SID, 4.4.4.4]
```

### Affinity Constraint Validation With Non-ECMP Anycast SID: Example

In this example, the shortest path to node 5 has an accumulative IGP metric of 20, and the shortest path to node 3 has an accumulative IGP metric of 30. Only the shortest path to node 5 is validated against affinity constraints.

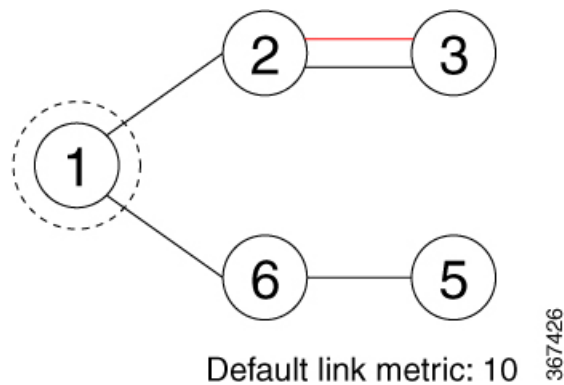


**Note** Even though parallel link (23) is marked with red, it is still considered valid since anycast traffic flows only on the path to node 5.

### Invalid Path Based on Affinity Constraint: Example

In this example, parallel link (23) is marked as red, so the path to anycast node 3 is invalidated.





```

SR-TE policy database
-----
Name: POLICY1 (Color: 2, End-point: 198.51.100.6)
Status:
  Admin: up Operational: up for 00:03:52 (since Jan 24 01:52:14.215)
Candidate-paths:
Preference 100:
  Constraints:
    Affinity:
      exclude-any: red
    Explicit: segment-list SIDLIST1 (inactive)
    Inactive Reason: Link [2.2.21.23,2.2.21.32] failed to satisfy affinity exclude-any
constraint=0x00000008, link attributes=0x0000000A

```

## Protocols

### Path Computation Element Protocol

The path computation element protocol (PCEP) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end label switched paths (LSPs) sourced from the PCC to a PCE peer. The PCE can request the PCC to update and modify parameters of LSPs it controls. The stateful model also enables a PCC to allow the PCE to initiate computations allowing the PCE to perform network-wide orchestration.

#### Configure the Head-End Router as PCEP PCC

Configure the head-end router as PCEP Path Computation Client (PCC) to establish a connection to the PCE. The PCC and PCE addresses must be routable so that TCP connection (to exchange PCEP messages) can be established between PCC and PCE.

##### Configure the PCC to Establish a Connection to the PCE

Use the **segment-routing traffic-eng pcc** command to configure the PCC source address, the SR-PCE address, and SR-PCE options.

A PCE can be given an optional precedence. If a PCC is connected to multiple PCEs, the PCC selects a PCE with the lowest precedence value. If there is a tie, a PCE with the highest IP address is chosen for computing path. The precedence *value* range is from 0 to 255.

```

Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# pcc
Router(config-sr-te-pcc)# source-address ipv4 local-source-address
Router(config-sr-te-pcc)# pce address ipv4 PCE-address[precedence value]
Router(config-sr-te-pcc)# pce address ipv4 PCE-address[keychain WORD]

```

### Configure PCEP-Related Timers

Use the **timers keepalive** command to specify how often keepalive messages are sent from PCC to its peers. The range is from 0 to 255 seconds; the default value is 30.

```
Router(config-sr-te-pcc)# timers keepalive seconds
```

Use the **timers deadtimer** command to specify how long the remote peers wait before bringing down the PCEP session if no PCEP messages are received from this PCC. The range is from 1 to 255 seconds; the default value is 120.

```
Router(config-sr-te-pcc)# timers deadtimer seconds
```

Use the **timers delegation-timeout** command to specify how long a delegated SR policy can remain up without an active connection to a PCE. The range is from 0 to 3600 seconds; the default value is 60.

```
Router(config-sr-te-pcc)# timers delegation-timeout seconds
```

### PCE-Initiated SR Policy Timers

Use the **timers initiated orphans** command to specify the amount of time that a PCE-initiated SR policy will remain delegated to a PCE peer that is no longer reachable by the PCC. The range is from 10 to 180 seconds; the default value is 180.

```
Router(config-sr-te-pcc)# timers initiated orphans seconds
```

Use the **timers initiated state** command to specify the amount of time that a PCE-initiated SR policy will remain programmed while not being delegated to any PCE. The range is from 15 to 14440 seconds (24 hours); the default value is 600.

```
Router(config-sr-te-pcc)# timers initiated state seconds
```

To better understand how the PCE-initiated SR policy timers operate, consider the following example:

- PCE A instantiates SR policy P at head-end N.
- Head-end N delegates SR policy P to PCE A and programs it in forwarding.
- If head-end N detects that PCE A is no longer reachable, then head-end N starts the PCE-initiated **orphan** and **state** timers for SR policy P.
- If PCE A reconnects before the **orphan** timer expires, then SR policy P is automatically delegated back to its original PCE (PCE A).
- After the **orphan** timer expires, SR policy P will be eligible for delegation to any other surviving PCE(s).
- If SR policy P is not delegated to another PCE before the **state** timer expires, then head-end N will remove SR policy P from its forwarding.

### Enable SR-TE SYSLOG Alarms

Use the **logging policy status** command to enable SR-TE related SYSLOG alarms.

```
Router(config-sr-te)# logging policy status
```

### Enable PCEP Reports to SR-PCE

Use the **report-all** command to enable the PCC to report all SR policies in its database to the PCE.

```
Router(config-sr-te-pcc)# report-all
```

### Customize MSD Value at PCC

Use the **maximum-sid-depth value** command to customize the Maximum SID Depth (MSD) signaled by PCC during PCEP session establishment.

The default MSD *value* is equal to the maximum MSD supported by the platform (10).

```
Router(config-sr-te)# maximum-sid-depth value
```

For cases with path computation at PCE, a PCC can signal its MSD to the PCE in the following ways:

- During PCEP session establishment – The signaled MSD is treated as a node-wide property.
  - MSD is configured under **segment-routing traffic-eng maximum-sid-depth value** command
- During PCEP LSP path request – The signaled MSD is treated as an LSP property.
  - On-demand (ODN) SR Policy: MSD is configured using the **segment-routing traffic-eng on-demand color color maximum-sid-depth value** command
  - Local SR Policy: MSD is configured using the **segment-routing traffic-eng policy WORD candidate-paths preference preference dynamic metric sid-limit value** command.



---

**Note** If the configured MSD values are different, the per-LSP MSD takes precedence over the per-node MSD.

---

After path computation, the resulting label stack size is verified against the MSD requirement.

- If the label stack size is larger than the MSD and path computation is performed by PCE, then the PCE returns a "no path" response to the PCC.
- If the label stack size is larger than the MSD and path computation is performed by PCC, then the PCC will not install the path.




---

**Note** A sub-optimal path (if one exists) that satisfies the MSD constraint could be computed in the following cases:

- For a dynamic path with TE metric, when the PCE is configured with the **pce segment-routing te-latency** command or the PCC is configured with the **segment-routing traffic-eng te-latency** command.
- For a dynamic path with LATENCY metric
- For a dynamic path with affinity constraints

For example, if the PCC MSD is 4 and the optimal path (with an accumulated metric of 100) requires 5 labels, but a sub-optimal path exists (with accumulated metric of 110) requiring 4 labels, then the sub-optimal path is installed.

---

### Customize the SR-TE Path Calculation

Use the **te-latency** command to enable ECMP-aware path computation for TE metric.

```
Router(config-sr-te) # te-latency
```




---

**Note** ECMP-aware path computation is enabled by default for IGP and LATENCY metrics.

---

### Configure PCEP Redundancy Type

Use the **redundancy pcc-centric** command to enable PCC-centric high-availability model. The PCC-centric model changes the default PCC delegation behavior to the following:

- After LSP creation, LSP is automatically delegated to the PCE that computed it.
- If this PCE is disconnected, then the LSP is redelegated to another PCE.
- If the original PCE is reconnected, then the delegation fallback timer is started. When the timer expires, the LSP is redelegated back to the original PCE, even if it has worse preference than the current PCE.

```
Router(config-sr-te-pcc) # redundancy pcc-centric
```

### Configuring Head-End Router as PCEP PCC and Customizing SR-TE Related Options: Example

The following example shows how to configure an SR-TE head-end router with the following functionality:

- Enable the SR-TE head-end router as a PCEP client (PCC) with 3 PCEP servers (PCE) with different precedence values. The PCE with IP address 10.1.1.57 is selected as BEST.
- Enable SR-TE related syslogs.
- Set the Maximum SID Depth (MSD) signaled during PCEP session establishment to 5.
- Enable PCEP reporting for all policies in the node.

```

segment-routing
traffic-eng
pcc
source-address ipv4 10.1.1.2
pce address ipv4 10.1.1.57
precedence 150
password clear <password>
!
pce address ipv4 10.1.1.58
precedence 200
password clear <password>
!
pce address ipv4 10.1.1.59
precedence 250
password clear <password>
!
!
logging
policy status
!
maximum-sid-depth 5
pcc
report-all
!
!
end

```

### Verification

```
RP/0/RSP0/CPU0:Router# show segment-routing traffic-eng pcc ipv4 peer
```

```
PCC's peer database:
```

```
-----
```

```
Peer address: 10.1.1.57, Precedence: 150, (best PCE)
```

```
State up
```

```
Capabilities: Stateful, Update, Segment-Routing, Instantiation
```

```
Peer address: 10.1.1.58, Precedence: 200
```

```
State up
```

```
Capabilities: Stateful, Update, Segment-Routing, Instantiation
```

```
Peer address: 10.1.1.59, Precedence: 250
```

```
State up
```

```
Capabilities: Stateful, Update, Segment-Routing, Instantiation
```

## BGP SR-TE

BGP may be used to distribute SR Policy candidate paths to an SR-TE head-end. Dedicated BGP SAFI and NLRI have been defined to advertise a candidate path of an SR Policy. The advertisement of Segment Routing policies in BGP is documented in the IETF draft <https://datatracker.ietf.org/doc/draft-ietf-idr-segment-routing-te-policy/>

SR policies with IPv4 and IPv6 end-points can be advertised over BGPv4 or BGPv6 sessions between the SR-TE controller and the SR-TE headend.

The Cisco IOS-XR implementation supports the following combinations:

- IPv4 SR policy advertised over BGPv4 session

- IPv6 SR policy advertised over BGPv4 session
- IPv6 SR policy advertised over BGPv6 session

## Configure BGP SR Policy Address Family at SR-TE Head-End

Perform this task to configure BGP SR policy address family at SR-TE head-end:

### SUMMARY STEPS

1. **configure**
2. **router bgp** *as-number*
3. **bgp router-id** *ip-address*
4. **address-family** {**ipv4** | **ipv6**} **sr-policy**
5. **exit**
6. **neighbor** *ip-address*
7. **remote-as** *as-number*
8. **address-family** {**ipv4** | **ipv6**} **sr-policy**
9. **route-policy** *route-policy-name* {**in** | **out**}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure</b>	
<b>Step 2</b>	<b>router bgp</b> <i>as-number</i>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config)# <b>router bgp</b> 65000	Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
<b>Step 3</b>	<b>bgp router-id</b> <i>ip-address</i>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config-bgp)# <b>bgp router-id</b> 10.1.1.1	Configures the local router with a specified router ID.
<b>Step 4</b>	<b>address-family</b> { <b>ipv4</b>   <b>ipv6</b> } <b>sr-policy</b>  <b>Example:</b>  RP/0/RSP0/CPU0:router(config-bgp)# <b>address-family</b> <b>ipv4 sr-policy</b>	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode.
<b>Step 5</b>	<b>exit</b>	

	Command or Action	Purpose
Step 6	<b>neighbor</b> <i>ip-address</i> <b>Example:</b> <pre>RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.10.0.1</pre>	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 7	<b>remote-as</b> <i>as-number</i> <b>Example:</b> <pre>RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 1</pre>	Creates a neighbor and assigns a remote autonomous system number to it.
Step 8	<b>address-family</b> { <i>ipv4</i>   <i>ipv6</i> } <b>sr-policy</b> <b>Example:</b> <pre>RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family ipv4 sr-policy</pre>	Specifies either the IPv4 or IPv6 address family and enters address family configuration submenu.
Step 9	<b>route-policy</b> <i>route-policy-name</i> { <b>in</b>   <b>out</b> } <b>Example:</b> <pre>RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# route-policy pass out</pre>	Applies the specified policy to IPv4 or IPv6 unicast routes.

#### Example: BGP SR-TE with BGPv4 Neighbor to BGP SR-TE Controller

The following configuration shows the an SR-TE head-end with a BGPv4 session towards a BGP SR-TE controller. This BGP session is used to signal both IPv4 and IPv6 SR policies.

```
router bgp 65000
bgp router-id 10.1.1.1
!
address-family ipv4 sr-policy
!
address-family ipv6 sr-policy
!
neighbor 10.1.3.1
remote-as 10
description *** eBGP session to BGP SRTE controller ***
address-family ipv4 sr-policy
route-policy pass in
route-policy pass out
!
address-family ipv6 sr-policy
route-policy pass in
route-policy pass out
!
!
```

#### Example: BGP SR-TE with BGPv6 Neighbor to BGP SR-TE Controller

The following configuration shows an SR-TE head-end with a BGPv6 session towards a BGP SR-TE controller. This BGP session is used to signal IPv6 SR policies.

```
router bgp 65000
  bgp router-id 10.1.1.1
  address-family ipv6 sr-policy
  !
  neighbor 3001::10:1:3:1
  remote-as 10
  description *** eBGP session to BGP SRTE controller ***
  address-family ipv6 sr-policy
    route-policy pass in
    route-policy pass out
  !
  !
  !
```

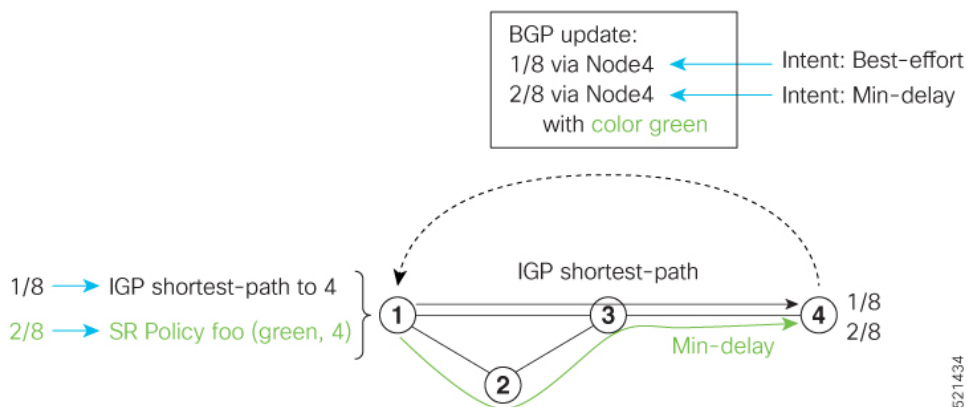
## Traffic Steering

### Automated Steering

Automated steering (AS) allows service traffic to be automatically steered onto the required transport SLA path programmed by an SR policy.

With AS, BGP automatically steers traffic onto an SR Policy based on the next-hop and color of a BGP service route. The color of a BGP service route is specified by a color extended community attribute. This color is used as a transport SLA indicator, such as min-delay or min-cost.

When the next-hop and color of a BGP service route matches the end-point and color of an SR Policy, BGP automatically installs the route resolving onto the BSID of the matching SR Policy. Recall that an SR Policy on a head-end is uniquely identified by an end-point and color.



When a BGP route has multiple extended-color communities, each with a valid SR Policy, the BGP process installs the route on the SR Policy giving preference to the color with the highest numerical value.

The granularity of AS behaviors can be applied at multiple levels, for example:

- At a service level—When traffic destined to all prefixes in a given service is associated to the same transport path type. All prefixes share the same color.



- At a destination/prefix level—When traffic destined to a prefix in a given service is associated to a specific transport path type. Each prefix could be assigned a different color.
- At a flow level—When flows destined to the same prefix are associated with different transport path types

AS behaviors apply regardless of the instantiation method of the SR policy, including:

- On-demand SR policy
- Manually provisioned SR policy
- PCE-initiated SR policy

See the [Verifying BGP VRF Information, on page 12](#) and [Verifying Forwarding \(CEF\) Table, on page 13](#) sections for sample output that shows AS implementation.

## Color-Only Automated Steering

Color-only steering is a traffic steering mechanism where a policy is created with given color, regardless of the endpoint.

You can create an SR-TE policy for a specific color that uses a NULL end-point (0.0.0.0 for IPv4 NULL, and ::0 for IPv6 NULL end-point). This means that you can have a single policy that can steer traffic that is based on that color and a NULL endpoint for routes with a particular color extended community, but different destinations (next-hop).



**Note** Every SR-TE policy with a NULL end-point must have an explicit path-option. The policy cannot have a dynamic path-option (where the path is computed by the head-end or PCE) since there is no destination for the policy.

You can also specify a color-only (CO) flag in the color extended community for overlay routes. The CO flag allows the selection of an SR-policy with a matching color, regardless of endpoint Sub-address Family Identifier (SAFI) (IPv4 or IPv6). See [Setting CO Flag, on page 58](#).

### Configure Color-Only Steering

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P1
Router(config-sr-te-policy)# color 1 end-point ipv4 0.0.0.0
```

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P2
Router(config-sr-te-policy)# color 2 end-point ipv6 ::0
```

```
Router# show running-configuration
segment-routing
traffic-eng
```

```

policy P1
  color 1 end-point ipv4 0.0.0.0
!
policy P2
  color 2 end-point ipv6 ::
!
!
!
end

```

## Setting CO Flag

The BGP-based steering mechanism matches BGP color and next-hop with that of an SR-TE policy. If the policy does not exist, BGP requests SR-PCE to create an SR-TE policy with the associated color, end-point, and explicit paths. For color-only steering (NULL end-point), you can configure a color-only (CO) flag as part of the color extended community in BGP.



**Note** See [Color-Only Automated Steering, on page 57](#) for information about color-only steering (NULL end-point).

The behavior of the steering mechanism is based on the following values of the CO flags:

<b>co-flag 00</b>	<ol style="list-style-type: none"> <li>1. The BGP next-hop and color &lt;N, C&gt; is matched with an SR-TE policy of same &lt;N, C&gt;.</li> <li>2. If a policy does not exist, then IGP path for the next-hop N is chosen.</li> </ol>
<b>co-flag 01</b>	<ol style="list-style-type: none"> <li>1. The BGP next-hop and color &lt;N, C&gt; is matched with an SR-TE policy of same &lt;N, C&gt;.</li> <li>2. If a policy does not exist, then an SR-TE policy with NULL end-point with the same address-family as N and color C is chosen.</li> <li>3. If a policy with NULL end-point with same address-family as N does not exist, then an SR-TE policy with any NULL end-point and color C is chosen.</li> <li>4. If no match is found, then IGP path for the next-hop N is chosen.</li> </ol>

### Configuration Example

```

Router(config)# extcommunity-set opaque overlay-color
Router(config-ext)# 1 co-flag 01
Router(config-ext)# end-set
Router(config)#
Router(config)# route-policy color
Router(config-rpl)# if destination in (5.5.5.1/32) then
Router(config-rpl-if)# set extcommunity color overlay-color
Router(config-rpl-if)# endif
Router(config-rpl)# pass
Router(config-rpl)# end-policy
Router(config)#

```

## Address-Family Agnostic Automated Steering

Address-family agnostic steering uses an SR-TE policy to steer both labeled and unlabeled IPv4 and IPv6 traffic. This feature requires support of IPv6 encapsulation (IPv6 caps) over IPV4 endpoint policy.

IPv6 caps for IPv4 NULL end-point is enabled automatically when the policy is created in Segment Routing Path Computation Element (SR-PCE). The binding SID (BSID) state notification for each policy contains an "ipv6\_caps" flag that notifies SR-PCE clients (PCC) of the status of IPv6 caps (enabled or disabled).

An SR-TE policy with a given color and IPv4 NULL end-point could have more than one candidate path. If any of the candidate paths has IPv6 caps enabled, then all of the remaining candidate paths need IPv6 caps enabled. If IPv6 caps is not enabled on all candidate paths of same color and end-point, traffic drops can occur.

You can disable IPv6 caps for a particular color and IPv4 NULL end-point using the **ipv6 disable** command on the local policy. This command disables IPv6 caps on all candidate paths that share the same color and IPv4 NULL end-point.

### Disable IPv6 Encapsulation

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# policy P1
Router(config-sr-te-policy)# color 1 end-point ipv4 0.0.0.0
Router(config-sr-te-policy)# ipv6 disable
```

## Using Binding Segments

The binding segment is a local segment identifying an SR-TE policy. Each SR-TE policy is associated with a binding segment ID (BSID). The BSID is a local label that is automatically allocated for each SR-TE policy when the SR-TE policy is instantiated.



**Note** In Cisco IOS XR 6.3.2 and later releases, you can specify an explicit BSID for an SR-TE policy. See the following **Explicit Binding SID** section.

BSID can be used to steer traffic into the SR-TE policy and across domain borders, creating seamless end-to-end inter-domain SR-TE policies. Each domain controls its local SR-TE policies; local SR-TE policies can be validated and rerouted if needed, independent from the remote domain's head-end. Using binding segments isolates the head-end from topology changes in the remote domain.

Packets received with a BSID as top label are steered into the SR-TE policy associated with the BSID. When the BSID label is popped, the SR-TE policy's SID list is pushed.

BSID can be used in the following cases:

- Multi-Domain (inter-domain, inter-autonomous system)—BSIDs can be used to steer traffic across domain borders, creating seamless end-to-end inter-domain SR-TE policies.
- Large-Scale within a single domain—The head-end can use hierarchical SR-TE policies by nesting the end-to-end (edge-to-edge) SR-TE policy within another layer of SR-TE policies (aggregation-to-aggregation). The SR-TE policies are nested within another layer of policies using the BSIDs, resulting in seamless end-to-end SR-TE policies.

- Label stack compression—If the label-stack size required for an SR-TE policy exceeds the platform capability, the SR-TE policy can be seamlessly stitched to, or nested within, other SR-TE policies using a binding segment.
- BGP SR-TE Dynamic—The head-end steers the packet into a BGP-based FIB entry whose next hop is a binding-SID.

### Explicit Binding SID

Use the **binding-sid mpls label** command in SR-TE policy configuration mode to specify the explicit BSID. Explicit BSIDs are allocated from the segment routing local block (SRLB) or the dynamic range of labels. A best-effort is made to request and obtain the BSID for the SR-TE policy. If requested BSID is not available (if it does not fall within the available SRLB or is already used by another application or SR-TE policy), the policy stays down.

Use the **binding-sid explicit {fallback-dynamic | enforce-srlb}** command to specify how the BSID allocation behaves if the BSID value is not available.

- Fallback to dynamic allocation – If the BSID is not available, the BSID is allocated dynamically and the policy comes up:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# binding-sid explicit fallback-dynamic
```

- Strict SRLB enforcement – If the BSID is not within the SRLB, the policy stays down:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# binding-sid explicit enforce-srlb
```

This example shows how to configure an SR policy to use an explicit BSID of 1000. If the BSID is not available, the BSID is allocated dynamically and the policy comes up.

```
segment-routing
traffic-eng
binding-sid explicit fallback-dynamic
policy goo
binding-sid mpls 1000
!
!
!
```

## Stitching SR-TE Polices Using Binding SID: Example

In this example, three SR-TE policies are stitched together to form a seamless end-to-end path from node 1 to node 10. The path is a chain of SR-TE policies stitched together using the binding-SIDs of intermediate policies, providing a seamless end-to-end path.



```

!
!
RP/0/RSP0/CPU0:Node-5# show segment-routing traffic-eng policy color 777

SR-TE policy database
-----

Color: 777, End-point: 10.1.1.10
Name: srte_c_777_ep_10.1.1.10
Status:
  Admin: up Operational: up for 00:00:52 (since Aug 19 07:40:12.662)
Candidate-paths:
  Preference: 100 (configuration) (active)
    Name: foo
    Requested BSID: 15888
    PCC info:
      Symbolic name: cfg_foo_discr_100
      PLSP-ID: 70
    Explicit: segment-list PATH-9_10 (valid)
      Weight: 1, Metric Type: TE
        16009 [Prefix-SID, 10.1.1.9]
        16010 [Prefix-SID, 10.1.1.10]
Attributes:
  Binding SID: 15888 (SRLB)
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

```

**Step 2** On node 3, do the following:

## a) Define an SR-TE policy with an explicit path configured using the following:

- Loopback interface IP address of node 4
- Interface IP address of link between node 4 and node 6
- Loopback interface IP address of node 5
- Binding-SID of the SR-TE policy defined in Step 1 (**mpls label 15888**)

**Note** This last segment allows the stitching of these policies.

b) Define an explicit binding-SID (**mpls label 15900**) allocated from SRLB for the SR-TE policy.**Example:****Node 3**

```

segment-routing
traffic-eng
segment-list PATH-4_4-6_5_BSID
index 10 address ipv4 10.1.1.4
index 20 address ipv4 10.4.6.6
index 30 address ipv4 10.1.1.5
index 40 mpls label 15888
!
policy baa
binding-sid mpls 15900
color 777 end-point ipv4 10.1.1.5
candidate-paths
preference 100
explicit segment-list PATH-4_4-6_5_BSID
!

```

```

!
!
!
!
RP/0/RSP0/CPU0:Node-3# show segment-routing traffic-eng policy color 777

SR-TE policy database
-----

Color: 777, End-point: 10.1.1.5
Name: srte_c_777_ep_10.1.1.5
Status:
  Admin: up Operational: up for 00:00:32 (since Aug 19 07:40:32.662)
Candidate-paths:
  Preference: 100 (configuration) (active)
  Name: baa
  Requested BSID: 15900
  PCC info:
    Symbolic name: cfg_baa_discr_100
    PLSP-ID: 70
  Explicit: segment-list PATH-4_4-6_5_BSID (valid)
  Weight: 1, Metric Type: TE
    16004 [Prefix-SID, 10.1.1.4]
    80005 [Adjacency-SID, 10.4.6.4 - 10.4.6.6]
    16005 [Prefix-SID, 10.1.1.5]
    15888
Attributes:
  Binding SID: 15900 (SRLB)
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

```

**Step 3** On node 1, define an SR-TE policy with an explicit path configured using the loopback interface IP address of node 3 and the binding-SID of the SR-TE policy defined in step 2 (**mpls label 15900**). This last segment allows the stitching of these policies.

### Example:

#### Node 1

```

segment-routing
traffic-eng
segment-list PATH-3_BSID
index 10 address ipv4 10.1.1.3
index 20 mpls label 15900
!
policy bar
color 777 end-point ipv4 10.1.1.3
candidate-paths
preference 100
explicit segment-list PATH-3_BSID
!
!
!
!
RP/0/RSP0/CPU0:Node-1# show segment-routing traffic-eng policy color 777

SR-TE policy database

```

```

-----
Color: 777, End-point: 10.1.1.3
Name: srte_c_777_ep_10.1.1.3
Status:
  Admin: up Operational: up for 00:00:12 (since Aug 19 07:40:52.662)
Candidate-paths:
  Preference: 100 (configuration) (active)
  Name: bar
  Requested BSID: dynamic
  PCC info:
    Symbolic name: cfg_bar_discr_100
    PLSP-ID: 70
  Explicit: segment-list PATH-3_BSID (valid)
  Weight: 1, Metric Type: TE
    16003 [Prefix-SID, 10.1.1.3]
    15900
Attributes:
  Binding SID: 80021
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

```

## L2VPN Preferred Path

EVPN VPWS Preferred Path over SR-TE Policy feature allows you to set the preferred path between the two end-points for EVPN VPWS pseudowire (PW) using SR-TE policy.

L2VPN VPLS or VPWS Preferred Path over SR-TE Policy feature allows you to set the preferred path between the two end-points for L2VPN Virtual Private LAN Service (VPLS) or Virtual Private Wire Service (VPWS) using SR-TE policy.

Refer to the [EVPN VPWS Preferred Path over SR-TE Policy](#) and [L2VPN VPLS or VPWS Preferred Path over SR-TE Policy](#) sections in the "L2VPN Services over Segment Routing for Traffic Engineering Policy" chapter of the *L2VPN and Ethernet Services Configuration Guide*.

## Static Route over Segment Routing Policy

This feature allows you to specify a Segment Routing (SR) policy as an interface type when configuring static routes for MPLS data planes.

For information on configuring static routes, see the "Implementing Static Routes" chapter in the *Routing Configuration Guide for Cisco ASR 9000 Series Routers*.

### Configuration Example

The following example depicts a configuration of a static route for an IPv4 destination over an SR policy according to following parameters:

- Target SR policy:
  - Color = 200
  - End-point = 10.1.1.4
  - Auto-generated SR policy name = srte\_c\_200\_ep\_10.1.1.4





**Note** Use the auto-generated SR-TE policy name to attach the SR policy to the static route. Auto-generated SR policy names use the following naming convention: **srte\_c\_color\_val\_ep\_endpoint-address**.

Use the `show segment-routing traffic-eng policy color <color_val> endpoint ipv4 <ip_addr>` command to display the auto-generated policy name.

- Admin distance = 40
- Load metric = 150
- Install the route in RIB regardless of reachability

```
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.1.1.4/32 sr-policy srte_c_200_ep_10.1.1.4 40 permanent metric
150
```

### Running Configuration

```
router static
 address-family ipv4 unicast
  10.1.1.4/32 sr-policy srte_c_200_ep_10.1.1.4 40 permanent metric 150
!
```

### Verification

```
RP/0/RP0/CPU0:RTR-1# show run segment-routing traffic-eng policy sample-policy-foo
Tue Feb 16 17:40:16.759 PST
```

```
segment-routing
 traffic-eng
  policy sample-policy-foo
   color 200 end-point ipv4 10.1.1.4
   candidate-paths
    preference 100
    dynamic
     metric
      type te
    !
  !
!
```

```
RP/0/RP0/CPU0:RTR-1# show segment-routing traffic-eng policy color 200 endpoint ipv4 10.1.1.4
Tue Feb 16 17:17:45.724 PST
```

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

```
Color: 200, End-point: 10.1.1.4
Name: srte_c_200_ep_10.1.1.4
```

```

Status:
  Admin: up Operational: up for 5d04h (since Feb 11 12:22:59.054)
Candidate-paths:
  Preference: 100 (configuration) (active)
  Name: sample-policy-foo
  Requested BSID: dynamic
  Protection Type: protected-preferred
  Maximum SID Depth: 10
  Dynamic (valid)
  Metric Type: TE, Path Accumulated Metric: 14
    16005 [Prefix-SID, 10.1.1.5]
    16004 [Prefix-SID, 10.1.1.4]
Attributes:
  Binding SID: 24014
  Forward Class: Not Configured
  Steering labeled-services disabled: no
  Steering BGP disabled: no
  IPv6 caps enable: yes
  Invalidation drop enabled: no

RP/0/RP0/CPU0:RTR-1# show static sr-policy srte_c_200_ep_10.1.1.4
Tue Feb 16 17:50:19.932 PST

Interface          VRF          State      Paths
srte_c_200_ep_10.1.1.4  default      Up         10.1.1.4/32
Reference Count(in path with both intf<-->NH):0
Last IM notification was Up at Feb 16 17:09:08.325

Global ifh          : 0x0000007c
IM state             : up
RSI registration    : Yes
Table IDs           : 0xe0000000

Address Info:
10.1.1.1/32
Route tag: 0x00000000 Flags: 0x00000000 Prefix SID: False [Active]

IP-STATIC-IDB-CLASS
Total entries : 1
Interface      : sr-srte_c_200_ep_10.1.1.4
| Event Name   | Time Stamp           | S, M
| idb-create   | Feb 16 17:09:08.352 | 0, 0

RP/0/RP0/CPU0:RTR-1# show route 10.1.1.4/32
Tue Feb 16 17:09:21.164 PST

Routing entry for 10.1.1.4/32
  Known via "static", distance 40, metric 0 (connected)
  Installed Feb 16 17:09:08.325 for 00:00:13
  Routing Descriptor Blocks
    directly connected, via srte_c_200_ep_10.1.1.4, permanent
    Route metric is 0, Wt is 150
  No advertising protos.

RP/0/RP0/CPU0:RTR-1# show route 10.1.1.4/32 detail
Tue Feb 16 17:09:36.718 PST

Routing entry for 10.1.1.4/32
  Known via "static", distance 40, metric 0 (connected)
  Installed Feb 16 17:09:08.325 for 00:00:28
  Routing Descriptor Blocks
    directly connected, via srte_c_200_ep_10.1.1.4, permanent
    Route metric is 0, Wt is 150

```

```

Label: None
Tunnel ID: None
Binding Label: None
Extended communities count: 0
NHID:0x0(Ref:0)
Route version is 0x4a (74)
Local Label: 0x3e84 (16004)
IP Precedence: Not Set
QoS Group ID: Not Set
Flow-tag: Not Set
Fwd-class: Not Set
Route Priority: RIB_PRIORITY_RECURSIVE (9) SVD Type RIB_SVD_TYPE_LOCAL
Download Priority 3, Download Version 258
No advertising protos.

```

```

RP/0/RP0/CPU0:RTR-1# show cef 10.1.1.4/32 detail
Tue Feb 16 17:10:06.956 PST
10.1.1.4/32, version 258, attached, internal 0x1000441 0x30 (ptr 0xd3f0d30) [1], 0x0
(0xe46f960), 0xa20 (0xe9694e0)
Updated Feb 16 17:09:08.328
Prefix Len 32, traffic index 0, precedence n/a, priority 3
gateway array (0xe2d9a08) reference count 2, flags 0x8068, source rib (7), 0 backups
[3 type 4 flags 0x108401 (0xe9aeb98) ext 0x0 (0x0)]
LW-LDI[type=1, refc=1, ptr=0xe46f960, sh-ldi=0xe9aeb98]
gateway array update type-time 1 Feb 16 17:07:59.946
LDI Update time Feb 16 17:07:59.946
LW-LDI-TS Feb 16 17:07:59.946
via srte_c_200_ep_10.1.1.4, 5 dependencies, weight 0, class 0 [flags 0xc]
path-idx 0 NHID 0x0 [0xf3b1a30 0x0]
local adjacency
local label 16004 labels imposed {None}

Load distribution: 0 (refcount 3)

Hash OK Interface Address
0 Y srte_c_200_ep_10.1.1.4 point2point

```

```

RP/0/RP0/CPU0:RTR-1# show mpls forwarding labels 16004 detail
Tue Feb 16 17:27:59.831 PST
Local Outgoing Prefix Outgoing Next Hop Bytes
Label Label or ID Interface Interface Switched
-----
16004 Unlabelled SR Pfx (idx 4) srte_c_200_e point2point 990
Updated: Feb 16 17:07:59.945
Path Flags: 0xc [ ]
Version: 258, Priority: 3
Label Stack (Top -> Bottom): { Unlabelled Unlabelled }
NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
MAC/Encaps: 0/0, MTU: 0
Outgoing Interface: srte_c_200_ep_10.1.1.4 (ifhandle 0x0000007c)
Packets Switched: 20

```

## Autoroute Include

You can configure SR-TE policies with Autoroute Include to steer specific IGP (IS-IS, OSPF) prefixes, or all prefixes, over non-shortest paths and to divert the traffic for those prefixes on to the SR-TE policy.

The **autoroute include all** option applies Autoroute Announce functionality for all destinations or prefixes.

The **autoroute include ipv4 address** option applies Autoroute Destination functionality for the specified destinations or prefixes. This option is supported for IS-IS only; it is not supported for OSPF.

The Autoroute SR-TE policy adds the prefixes into the IGP, which determines if the prefixes on the endpoint or downstream of the endpoint are eligible to use the SR-TE policy. If a prefix is eligible, then the IGP checks if the prefix is listed in the Autoroute Include configuration. If the prefix is included, then the IGP downloads the prefix route with the SR-TE policy as the outgoing path.

### Usage Guidelines and Limitations

- Autoroute Include supports three metric types:
  - Default (no metric): The path over the SR-TE policy inherits the shortest path metric.
  - Absolute (constant) metric: The shortest path metric to the policy endpoint is replaced with the configured absolute metric. The metric to any prefix that is Autoroute Included is modified to the absolute metric. Use the **autoroute metric constant** *constant-metric* command, where *constant-metric* is from 1 to 2147483647.
  - Relative metric: The shortest path metric to the policy endpoint is modified with the relative value configured (plus or minus). Use the **autoroute metric relative** *relative-metric* command, where *relative-metric* is from -10 to +10.




---

**Note** To prevent load-balancing over IGP paths, you can specify a metric that is lower than the value that IGP takes into account for autorouted destinations (for example, **autoroute metric relative -1**).

---

### Configuration Examples

The following example shows how to configure autoroute include for all prefixes:

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)#policy P1
Router(config-sr-te-policy)# color 20 end-point ipv4 10.1.1.2
Router(config-sr-te-policy)# autoroute include all
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-pp-index)# explicit segment-list Plist-1
```

The following example shows how to configure autoroute include for the specified IPv4 prefixes:




---

**Note** This option is supported for IS-IS only; it is not supported for OSPF.

---

```
Router# configure
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)#policy P1
Router(config-sr-te-policy)# color 20 end-point ipv4 10.1.1.2
Router(config-sr-te-policy)# autoroute include ipv4 10.1.1.21/32
Router(config-sr-te-policy)# autoroute include ipv4 10.1.1.23/32
```

```
Router(config-sr-te-policy)# autoroute metric constant 1
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-pp-index)# explicit segment-list Plist-1
```

## Policy-Based Tunnel Selection for SR-TE Policy

Policy-Based Tunnel Selection (PBTS) is a mechanism that lets you direct traffic into specific SR-TE policies based on different classification criteria. PBTS benefits Internet service providers (ISPs) that carry voice and data traffic through their networks, who want to route this traffic to provide optimized voice service.

PBTS works by selecting SR-TE policies based on the classification criteria of the incoming packets, which are based on the IP precedence, experimental (EXP), differentiated services code point (DSCP), or type of service (ToS) field in the packet. Default-class configured for paths is always zero (0). If there is no TE for a given forward-class, then the default-class (0) will be tried. If there is no default-class, then the packet is dropped. PBTS supports up to seven (exp 1 - 7) EXP values associated with a single SR-TE policy.

For more information about PBTS, refer to the "Policy-Based Tunnel Selection" section in the *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*.

### Configure Policy-Based Tunnel Selection for SR-TE Policies

The following section lists the steps to configure PBTS for an SR-TE policy.



**Note** Steps 1 through 4 are detailed in the "Implementing MPLS Traffic Engineering" chapter of the *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*.

1. Define a class-map based on a classification criteria.
2. Define a policy-map by creating rules for the classified traffic.
3. Associate a forward-class to each type of ingress traffic.
4. Enable PBTS on the ingress interface, by applying this service-policy.
5. Create one or more egress SR-TE policies (to carry packets based on priority) to the destination and associate the egress SR-TE policy to a forward-class.

### Configuration Example

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy POLICY-PBTS
Router(config-sr-te-policy)# color 1001 end-point ipv4 10.1.1.20
Router(config-sr-te-policy)# autoroute
Router(config-sr-te-policy-autoroute)# include all
Router(config-sr-te-policy-autoroute)# forward-class 1
Router(config-sr-te-policy-autoroute)# exit
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 1
Router(config-sr-te-policy-path-pref)# explicit segment-list SIDLIST1
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-pp-info)# exit
Router(config-sr-te-policy-path-pref)# exit
Router(config-sr-te-policy-path)# preference 2
```

```
Router(config-sr-te-policy-path-pref)# dynamic
Router(config-sr-te-pp-info)# metric
Router(config-sr-te-path-metric)# type te
Router(config-sr-te-path-metric)# commit
```

### Running Configuration

```
segment-routing
traffic-eng
policy POLICY-PBTS
color 1001 end-point ipv4 10.1.1.20
autoroute
include all
forward-class 1
!
candidate-paths
preference 1
explicit segment-list SIDLIST1
!
!
preference 2
dynamic
metric
type te
```

## Miscellaneous

### LDP over Segment Routing Policy

The LDP over Segment Routing Policy feature enables an LDP-targeted adjacency over a Segment Routing (SR) policy between two routers. This feature extends the existing MPLS LDP address family neighbor configuration to specify an SR policy as the targeted end-point.

LDP over SR policy is supported for locally configured SR policies with IPv4 end-points.

For more information about MPLS LDP, see the "Implementing MPLS Label Distribution Protocol" chapter in the *MPLS Configuration Guide*.

For more information about Autoroute, see the *Autoroute Announce for SR-TE* section.




---

**Note** Before you configure an LDP targeted adjacency over SR policy name, you need to create the SR policy under Segment Routing configuration. The SR policy interface names are created internally based on the color and endpoint of the policy. LDP is non-operational if SR policy name is unknown.

---

The following functionality applies:

1. Configure the SR policy – LDP receives the associated end-point address from the interface manager (IM) and stores it in the LDP interface database (IDB) for the configured SR policy.
2. Configure the SR policy name under LDP – LDP retrieves the stored end-point address from the IDB and uses it. Use the auto-generated SR policy name assigned by the router when creating an LDP targeted

adjacency over an SR policy. Auto-generated SR policy names use the following naming convention: `srte_c_color_val_ep_endpoint-address`. For example, `srte_c_1000_ep_10.1.1.2`

### Configuration Example

```
/* Enter the SR-TE configuration mode and create the SR policy. This example corresponds
to a local SR policy with an explicit path. */
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# segment-list sample-sid-list
Router(config-sr-te-sl)# index 10 address ipv4 10.1.1.7
Router(config-sr-te-sl)# index 20 address ipv4 10.1.1.2
Router(config-sr-te-sl)# exit
Router(config-sr-te)# policy sample_policy
Router(config-sr-te-policy)# color 1000 end-point ipv4 10.1.1.2
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list sample-sid-list
Router(config-sr-te-pp-info)# end

/* Configure LDP over an SR policy */
Router(config)# mpls ldp
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# neighbor sr-policy srte_c_1000_ep_10.1.1.2 targeted
Router(config-ldp-af)#
```



**Note** Do one of the following to configure LDP discovery for targeted hellos:

- Active targeted hellos (SR policy head end):

```
mpls ldp
  interface GigabitEthernet0/0/0/0
  !
  !
```

- Passive targeted hellos (SR policy end-point):

```
mpls ldp
  address-family ipv4
    discovery targeted-hello accept
  !
  !
```

### Running Configuration

```
segment-routing
  traffic-eng
    segment-list sample-sid-list
      index 10 address ipv4 10.1.1.7
      index 20 address ipv4 10.1.1.2
    !
  policy sample_policy
    color 1000 end-point ipv4 10.1.1.2
    candidate-paths
      preference 100
      explicit segment-list sample-sid-list
    !
```

```

!
!
!
!
!
mpls ldp
address-family ipv4
neighbor sr-policy srte_c_1000_ep_10.1.1.2 targeted
  discovery targeted-hello accept
!
!

```

## Verification

```

Router# show mpls ldp interface brief

```

Interface	VRF Name	Config	Enabled	IGP-Auto-Cfg	TE-Mesh-Grp	cfg
Te0/3/0/0/3	default	Y	Y	0	N/A	
Te0/3/0/0/6	default	Y	Y	0	N/A	
Te0/3/0/0/7	default	Y	Y	0	N/A	
Te0/3/0/0/8	default	N	N	0	N/A	
Te0/3/0/0/9	default	N	N	0	N/A	
<b>srte_c_1000_</b>	default	<b>Y</b>	<b>Y</b>	0	N/A	

```

Router# show mpls ldp interface
Interface TenGigE0/3/0/0/3 (0xa000340)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface
Interface TenGigE0/3/0/0/6 (0xa000400)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface
Interface TenGigE0/3/0/0/7 (0xa000440)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface
Interface TenGigE0/3/0/0/8 (0xa000480)
  VRF: 'default' (0x60000000)
  Disabled:
Interface TenGigE0/3/0/0/9 (0xa0004c0)
  VRF: 'default' (0x60000000)
  Disabled:
Interface srte_c_1000_ep_10.1.1.2 (0x520)
  VRF: 'default' (0x60000000)
  Enabled via config: LDP interface

```

```

Router# show segment-routing traffic-eng policy color 1000

```

```

SR-TE policy database
-----

```

```

Color: 1000, End-point: 10.1.1.2
Name: srte_c_1000_ep_10.1.1.2
Status:
  Admin: up Operational: up for 00:02:00 (since Jul  2 22:39:06.663)
Candidate-paths:
  Preference: 100 (configuration) (active)
  Name: sample_policy
  Requested BSID: dynamic
  PCC info:
    Symbolic name: cfg_sample_policy_discr_100
    PLSP-ID: 17
  Explicit: segment-list sample-sid-list (valid)

```



```

Weight: 1, Metric Type: TE
  16007 [Prefix-SID, 10.1.1.7]
  16002 [Prefix-SID, 10.1.1.2]
Attributes:
  Binding SID: 80011
  Forward Class: 0
  Steering BGP disabled: no
  IPv6 caps enable: yes

```

```
Router# show mpls ldp neighbor 10.1.1.2 detail
```

```

Peer LDP Identifier: 10.1.1.2:0
TCP connection: 10.1.1.2:646 - 10.1.1.6:57473
Graceful Restart: No
Session Holdtime: 180 sec
State: Oper; Msgs sent/rcvd: 421/423; Downstream-Unsolicited
Up time: 05:22:02
LDP Discovery Sources:
  IPv4: (1)
    Targeted Hello (10.1.1.6 -> 10.1.1.2, active/passive)
  IPv6: (0)
Addresses bound to this peer:
  IPv4: (9)
    10.1.1.2      2.2.2.99      10.1.2.2      10.2.3.2
    10.2.4.2      10.2.22.2     10.2.222.2    10.30.110.132
    11.2.9.2
  IPv6: (0)
Peer holdtime: 180 sec; KA interval: 60 sec; Peer state: Estab
NSR: Disabled
Clients: LDP over SR Policy
Capabilities:
  Sent:
    0x508 (MP: Point-to-Multipoint (P2MP))
    0x509 (MP: Multipoint-to-Multipoint (MP2MP))
    0x50a (MP: Make-Before-Break (MBB))
    0x50b (Typed Wildcard FEC)
  Received:
    0x508 (MP: Point-to-Multipoint (P2MP))
    0x509 (MP: Multipoint-to-Multipoint (MP2MP))
    0x50a (MP: Make-Before-Break (MBB))
    0x50b (Typed Wildcard FEC)

```

## SR-TE Reoptimization Timers

SR-TE path re-optimization occurs when the head-end determines that there is a more optimal path available than the one currently used. For example, in case of a failure along the SR-TE LSP path, the head-end could detect and revert to a more optimal path by triggering re-optimization.

Re-optimization can occur due to the following events:

- The explicit path hops used by the primary SR-TE LSP explicit path are modified
- The head-end determines the currently used path-option are invalid due to either a topology path disconnect, or a missing SID in the SID database that is specified in the explicit-path
- A more favorable path-option (lower index) becomes available

For event-based re-optimization, you can specify various delay timers for path re-optimization. For example, you can specify how long to wait before switching to a reoptimized path

Additionally, you can configure a timer to specify how often to perform reoptimization of policies. You can also trigger an immediate reoptimization for a specific policy or for all policies.

### SR-TE Reoptimization

To trigger an immediate SR-TE reoptimization, use the **segment-routing traffic-eng reoptimization** command in Exec mode:

```
Router# segment-routing traffic-eng reoptimization {all | name policy}
```

Use the **all** option to trigger an immediate reoptimization for all policies. Use the **name policy** option to trigger an immediate reoptimization for a specific policy.

### Configuring SR-TE Reoptimization Timers

Use these commands in SR-TE configuration mode to configure SR-TE reoptimization timers:

- **timers candidate-path cleanup-delay seconds**—Specifies the delay before cleaning up candidate paths, in seconds. The range is from 0 (immediate clean-up) to 86400; the default value is 120
- **timers cleanup-delay seconds**—Specifies the delay before cleaning up previous path, in seconds. The range is from 0 (immediate clean-up) to 300; the default value is 10.
- **timers init-verify-restart seconds**—Specifies the delay for topology convergence after the topology starts populating due to a restart, in seconds. The range is from 10 to 10000; the default is 40.
- **timers init-verify-startup seconds**—Specifies the delay for topology convergence after topology starts populating for due to startup, in seconds. The range is from 10 to 10000; the default is 300
- **timers init-verify-switchover seconds**—Specifies the delay for topology convergence after topology starts populating due to a switchover, in seconds. The range is from 10 to 10000; the default is 60.
- **timers install-delay seconds**—Specifies the delay before switching to a reoptimized path, in seconds. The range is from 0 (immediate installation of new path) to 300; the default is 10.
- **timers periodic-reoptimization seconds**—Specifies how often to perform periodic reoptimization of policies, in seconds. The range is from 0 to 86400; the default is 600.

### Example Configuration

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# timers
Router(config-sr-te-timers)# candidate-path cleanup-delay 600
Router(config-sr-te-timers)# cleanup-delay 60
Router(config-sr-te-timers)# init-verify-restart 120
Router(config-sr-te-timers)# init-verify-startup 600
Router(config-sr-te-timers)# init-verify-switchover 30
Router(config-sr-te-timers)# install-delay 60
Router(config-sr-te-timers)# periodic-reoptimization 3000
```

### Running Config

```
segment-routing
 traffic-eng
  timers
   install-delay 60
   periodic-reoptimization 3000
   cleanup-delay 60
```

```
candidate-path cleanup-delay 600
init-verify-restart 120
init-verify-startup 600
init-verify-switchover 30
!
!
!
```

