SRv6-TE Overview

Traffic engineering over SRv6 can be accomplished in the following ways:

- End-to-End Flexible Algorithm: This is used for traffic engineering intents achieved with Flexible Algorithm, including low latency, multi-plane disjointness, affinity inclusion/exclusion, and SRLG exclusion.
- SRv6-TE Policy: This is used for traffic engineering intents beyond Flex Algo capabilities, such as path disjointness that rely on path computation by a PCE. In addition, this is used for user-configured explicit paths.

End-to-End Flexible Algorithm

SRv6 Flexible Algorithm allows operators to customize IGP shortest path computation according to their own needs. An operator can assign custom SRv6 locators to realize forwarding beyond link-cost-based shortest path. As a result, Flexible Algorithm provides a traffic-engineered path automatically computed by the IGP to any destination reachable by the IGP.

SRv6-TE Policy

SRv6 for traffic engineering (SRv6-TE) uses a “policy” to steer traffic through the network. An SRv6-TE policy path is expressed as a list of micro-segments that specifies the path, called a micro-segment ID (uSID) list. Each segment list 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 SRv6-TE policy, the uSID list is pushed on the packet by the head-end. The rest of the network executes the instructions embedded in the uSID list.

An SRv6-TE policy is identified as an ordered list (head-end, color, end-point):

- Head-end – Where the SRv6-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 SRv6-TE policy

Every SRv6-TE policy has a color value. Every policy between the same node pairs requires a unique color value.

An SRv6-TE policy uses one or more candidate paths. A candidate path can be made up of a single SID-list or a set of weighted SID-lists (for weighted equal cost multi-path [WECMP]).

A SID list can be either the result of a dynamic path computation by a PCE or a user-configured explicit path. See SRv6-TE Policy Path Types for more information.

Usage Guidelines and Limitations

Observe the following usage guidelines and limitations for the platform.

This release supports the following SRv6 policy functionality:

- SRv6 policies with SRv6 uSID segments
- SRv6 policies with PCE-delegated dynamic paths
- SRv6 policies with explicit paths
- Path delegation and reporting with PCEPv4
- Path delegation and reporting with PCEPv6
- PCEPv4 and PCEPv6 sessions with different PCEs
- PCEP path delegation with the following optimization objective (metric) types:
  - IGP metric
  - TE metric
  - Delay (latency)
  - Hop-count
- PCEP path delegation with the following constraint types:
  - Affinity (include-any, include-all, exclude-any)
  - Path disjointness (link, node, SRLG, SRLG+node)
- Steering over SRv6 policies with Automated Steering for the following services:
  - L3 BGP-based services (IPv4 L3VPN, IPv6 L3VPN, IPv4 BGP global, IPv6 BGP global)

The following functionalities are not supported:

- SRv6 policy counters
- PCE-initiated SRv6 policies via PCEP
- SRv6 policies with head-end computed dynamic paths
- PCE path delegation with segment-type Flex Algo constraint
- PCEPv4 and PCEPv6 sessions with same PCE

- Steering over SRv6 policies based on incoming BSID (remote automated steering)
- PCC with user-configured PCE groups
- SR-PM delay-measurement over SRv6 policies
- SR-PM liveness detection over SRv6 policies
- L3 services with BGP PIC over SRv6 policies
- SRv6 policy ping

SR-PCE and SRv6-TE head-end PCEP compatibility:

- A PCEP session between an SR-PCE running Cisco IOS XR release 7.8.1 and an SRv6-TE head-end running Cisco IOS XR release 7.8.1 is supported.
- A PCEP session between an SR-PCE running Cisco IOS XR release 7.8.1 and an SRv6-TE head-end running an earlier release is supported.
- A PCEP session between an SR-PCE running Cisco IOS XR release earlier than 7.8.1 and an SRv6-TE head-end running Cisco IOS XR release 7.8.1 is not supported.
- A PCEP session between an SR-PCE running Cisco IOS XR release 7.8.1 and another SR-PCE running an earlier Cisco IOS XR release is not supported.
- As a result of the above, prior to upgrading any SRv6-TE headend nodes in the network to Cisco IOS XR release 7.8.1, you must upgrade its SR-PCE to Cisco IOS XR release 7.8.1.

Note When a pair of SR-PCEs part of the same redundancy group run different versions of Cisco IOS XR, they will lose their state-sync PCEP session until both SR-PCEs are upgraded to Cisco IOS XR release 7.8.1.

Instantiation of an SRv6 Policy

An SRv6 policy is instantiated, or implemented, at the head-end router. An SRv6 policy can be instantiated in the following ways:

- Manually provisioned SRv6 policy
- SR On-Demand Next-Hop (SR-ODN)
- PCE-initiated

The following sections provide details on the supported SRv6 policy instantiation methods:

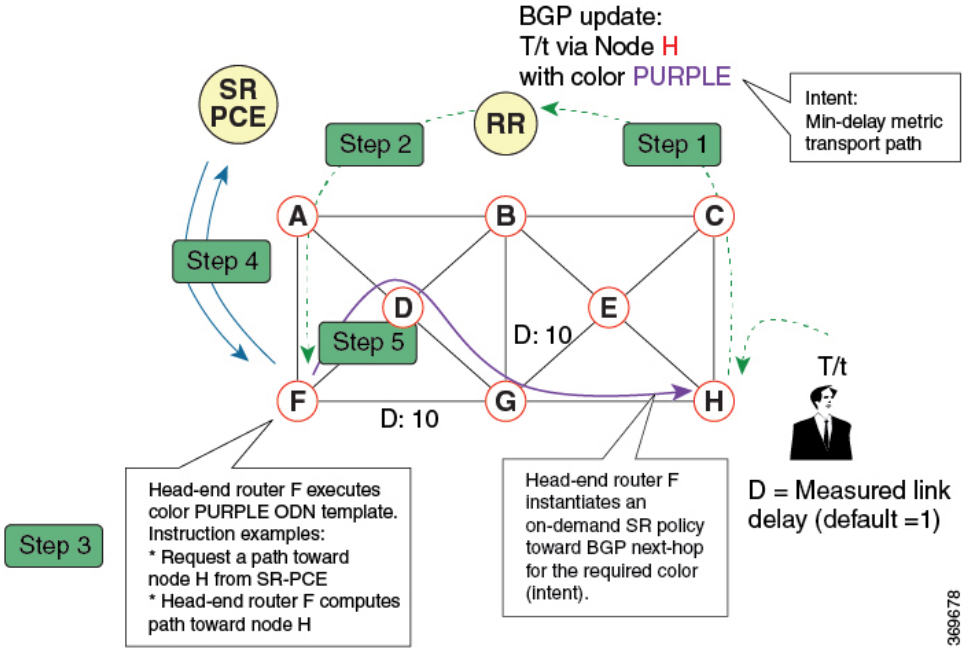
- Manually Provisioned SRv6 Policy, on page 18

On-Demand SRv6 Policy – SR On-Demand Next-Hop

SR On-Demand Next Hop (SR-ODN) allows a service head-end router to automatically instantiate an SRv6 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 SRv6 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 SRv6 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 SRv6 policy. When the last prefix associated with a given [intent, egress PE] pair is withdrawn, the on-demand SRv6 policy is deleted, and resources are freed from the head-end router.

An on-demand SRv6 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)

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 *BGP Configuration Guide for Cisco NCS 560 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 NCS 560 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

Attach Point	Set	Match
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 NCS 560 Series Routers*.

Configure On-Demand Color Template

```
Placeholder for "locator" config, need to capture inheritance rules

RP/0/RSP0/CPU0:ios(config-sr)# show parser dump | inc locator
srv6 locators locator WORD
srv6 locators locator WORD micro-segment behavior unode shift-only
srv6 locators locator WORD micro-segment behavior unode psp-usd
srv6 locators locator WORD prefix <IPv4/v6 Addr>
srv6 locators locator WORD algorithm <128-255>
srv6 locators locator WORD anycast
srv6 locators locator WORD delayed-delete

traffic-eng srv6 locator WORD binding-sid dynamic behavior ub6-insert-reduced
traffic-eng policy WORD srv6 locator WORD binding-sid dynamic behavior ub6-insert-reduced
traffic-eng on-demand color <1-4294967295> srv6 locator WORD binding-sid dynamic behavior
ub6-insert-reduced

segment-routing traffic-eng [policy WORD | on-demand color color] srv6 locator WORD
binding-sid dynamic behavior ub6-insert-reduced

   color range = 1 - 4294967295

Ex:
segment-routing srv6 locators locator WORD
segment-routing traffic-eng srv6 locator WORD binding-sid dynamic behavior ub6-insert-reduced

segment-routing traffic-eng policy WORD srv6 locator WORD binding-sid dynamic behavior
ub6-insert-reduced
segment-routing traffic-eng on-demand color color srv6 locator WORD binding-sid dynamic
behavior ub6-insert-reduced
```

Configure SRv6 ODN Color Template and Binding SID Behavior

- Use the **segment-routing traffic-eng 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.

  Use the **srv6** keyword to enable SRv6 forwarding.

  You can configure a per-ODN locator and binding SID (BSID) behavior, or use the global locator. To configure a per-ODN locator and BSID behavior, use the **locator** *locator* **binding-sid dynamic behavior ub6-insert-reduced** command.

  ```
  Router(config)# segment-routing traffic-eng
  Router(config-sr-te)# on-demand color 20
  Router(config-sr-te-color)# srv6
  Router(config-sr-te-color-srv6)# locator MyODNLocator binding-sid dynamic behavior
  ub6-insert-reduced
  Router(config-sr-te-color-srv6)#
  ```

Running Config

```
segment-routing
 traffic-eng
  on-demand color 20
   srv6
    locator MyODNLocator binding-sid dynamic behavior ub6-insert-reduced
   !
  !
 !
!
```

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 **segment-routing traffic-eng 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.

  Use the **srv6** keyword to enable SRv6 forwarding.

  You can configure a per-ODN locator and binding SID (BSID) behavior, or use the global locator. To configure a per-ODN locator and BSID behavior, use the **locator** *locator* **binding-sid dynamic behavior ub6-insert-reduced** command.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 20
Router(config-sr-te-color)# dynamic pcep
Router(config-sr-te-color-dyn-pce)# exit
Router(config-sr-te-color-dyn)# exit
Router(config-sr-te-color)# srv6
Router(config-sr-te-color-srv6) locator MyODNLocator-pcep binding-sid dynamic behavior
 ub6-insert-reduced
Router(config-sr-te-color-srv6) exit
```

Running Config

```
segment-routing
 traffic-eng
  on-demand color 20
   srv6
    locator MyODNLocator-pcep binding-sid dynamic behavior ub6-insert-reduced
   !
   dynamic
    pcep
    !
   !
  !
 !
!
```

Configure Dynamic Path Optimization Objectives

- Use the **metric type** {**hopcount** | **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

Note Disjoint-path and affinity constraints cannot be configured at the same time.

- 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 brown
```

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

  The default MSD *value* is equal to the maximum MSD supported by the platform (6 — 1 carrier, 6 uSIDs per carrier).

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

  The MSD value is not used during path computation. The MSD value is used as a pass or fail criteria after path computation.

  See Customize MSD Value at PCC, on page 29 for information about SR-TE label imposition capabilities.

- Use the **constraints segments protection** {**protected-only** | **protected-preferred** | **unprotected-only** | **unprotected-preferred**} command to configure the segment protection-type behavior.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# on-demand color 20
Router(config-sr-te-color)# constraints
Router(config-sr-te-color-const)# segments
Router(config-sr-te-color-const-seg)# protection protected-only
```

Running Config

```
segment-routing
 traffic-eng
  on-demand color 20
   srv6
    locator MyODNLocator-pcep binding-sid dynamic behavior ub6-insert-reduced
   !
   dynamic
    pcep
    !
    metric
     type te
     margin absolute 5
```

```
    !
    affinity
     exclude-any
      name brown
     !
    !
   !
   maximum-sid-depth 5
  !
 !
!
```

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 SRv6-TE head-end nodes. The following example shows various ODN color templates:

- color 20: minimization objective = te-metric; path computation at SR-PCE
- color 30: minimization objective = delay-metric; path computation at SR-PCE
- color 40: minimization objective = igp-metric; path computation at SR-PCE
- color 50: minimization objective = igp-metric; path computation at SR-PCE; constraints = affinity

Configure the global SRv6 locator and BSID behavior:

```
Node1(config)# segment-routing
Node1(config-sr)# traffic-eng
Node1(config-sr-te)# srv6
Node1(config-sr-te-srv6)# locator Node1 binding-sid dynamic behavior ub6-insert-reduced
Node1(config-sr-te-srv6)# exit
```

Configure ODN color template 20 with TE metric:

```
Node1(config-sr-te)# on-demand color 20
Node1(config-sr-te-color)# srv6
Node1(config-sr-te-color-srv6)# exit
Node1(config-sr-te-color)# dynamic pcep
Node1(config-sr-te-color-dyn-pce)# exit
Node1(config-sr-te-color-dyn)# metric type te
Node1(config-sr-te-color-dyn)# exit
Node1(config-sr-te-color)# exit
Node1(config-sr-te)#
```

Configure ODN color template 30 with delay metric:

```
Node1(config-sr-te)# on-demand color 30
Node1(config-sr-te-color)# srv6
Node1(config-sr-te-color-srv6)# exit
Node1(config-sr-te-color)# dynamic pcep
Node1(config-sr-te-color-dyn-pce)# exit
Node1(config-sr-te-color-dyn)# metric type latency
Node1(config-sr-te-color-dyn)# exit
Node1(config-sr-te-color)# exit
```

```
Node1(config-sr-te)#
```

Configure ODN color template 40 with IGP metric:

```
Node1(config-sr-te)# on-demand color 40
Node1(config-sr-te-color)# srv6
Node1(config-sr-te-color-srv6)# exit
Node1(config-sr-te-color)# dynamic pcep
Node1(config-sr-te-color-dyn-pce)# exit
Node1(config-sr-te-color-dyn)# metric type igp
Node1(config-sr-te-color-dyn)# exit
Node1(config-sr-te-color)# exit
Node1(config-sr-te)#
```

Configure ODN color template 50 with IGP metric and affinity constraint:

```
Node1(config-sr-te)# on-demand color 50
Node1(config-sr-te-color)# srv6
Node1(config-sr-te-color-srv6)# exit
Node1(config-sr-te-color)# dynamic pcep
Node1(config-sr-te-color-dyn-pce)# exit
Node1(config-sr-te-color-dyn)# metric type igp
Node1(config-sr-te-color-dyn)# affinity exclude-any name brown
Node1(config-sr-te-color-dyn)# commit
```

Running Config

```
segment-routing
 traffic-eng
  srv6
   locator Node1 binding-sid dynamic behavior ub6-insert-reduced
  !
  on-demand color 20
   dynamic
    pcep
    !
    metric
     type te
    !
   !
  !
  on-demand color 30
   srv6
   !
   dynamic
    pcep
    !
    metric
     type latency
    !
   !
  !
  on-demand color 40
   srv6
   !
   dynamic
    pcep
    !
    metric
     type igp
    !
   !
```

```
  !
  on-demand color 50
   srv6
   !
   dynamic
    pcep
    !
    metric
     type igp
    !
    affinity
     exclude-any
      name brown
     !
    !
   !
  !
 !
!
```

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.

```
Router(config)# extcommunity-set opaque color20-te
Router(config-ext)# 20
Router(config-ext)# end-set

Router(config)# extcommunity-set opaque color30-delay
Router(config-ext)# 30
Router(config-ext)# end-set

Router(config)# extcommunity-set opaque color40-igp
Router(config-ext)# 40
Router(config-ext)# end-set

Router(config)# extcommunity-set opaque color50-igp-excl-brown
Router(config-ext)# 50
Router(config-ext)# end-set
```

Running Config

```
extcommunity-set opaque color20-te
  20
end-set
!
extcommunity-set opaque color30-delay
  30
end-set
!
extcommunity-set opaque color40-igp
  40
end-set
!
```

```
extcommunity-set opaque color50-igp-exclude-brown
  50
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.

```
Node1(config)# route-policy SET_COLOR_LOW_TE
Node1(config-rpl)# set extcommunity color color20-te
Node1(config-rpl)# pass
Node1(config-rpl)# end-policy

Node1(config)# route-policy SET_COLOR_LOW_LATENCY
Node1(config-rpl)# set extcommunity color color30-delay
Node1(config-rpl)# pass
Node1(config-rpl)# end-policy

Node1(config)# route-policy SET_COLOR_HI_BW
Node1(config-rpl)# set extcommunity color color40-igp
Node1(config-rpl)# pass
Node1(config-rpl)# end-policy

Node1(config)# route-policy SET_COLOR_HI_BW_exclude-brown
Node1(config-rpl)# set extcommunity color color50-igp-exclude-brown
Node1(config-rpl)# pass
Node1(config-rpl)#end-policy
```

Running Config

```
route-policy SET_COLOR_LOW_TE
  set extcommunity color color20-te
  pass
end-policy
!
route-policy SET_COLOR_LOW_LATENCY
  set extcommunity color color30-delay
  pass
end-policy
!
route-policy SET_COLOR_HI_BW
  set extcommunity color color40-igp
  pass
end-policy
!
route-policy SET_COLOR_HI_BW_exclude-brown
  set extcommunity color color50-igp-exclude-brown
  pass
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.

```
vrf vrf_cust1
 address-family ipv4 unicast
```

```
  import route-target
   100:1
  !
  export route-policy SET_COLOR_LOW_TE
  export route-target
   1:1
   100:1
  !
 !
!
vrf vrf_cust2
 address-family ipv4 unicast
  import route-target
   100:2
  !
  export route-policy SET_COLOR_LOW_LATENCY
  export route-target
   100:2
  !
 !
!
vrf vrf_cust3
 address-family ipv4 unicast
  import route-target
   100:3
  !
  export route-policy SET_COLOR_HI_BW
  export route-target
   100:3
  !
 !
!
vrf vrf_cust4
 address-family ipv4 unicast
  import route-target
   100:4
  !
  export route-policy SET_COLOR_HI_BW_exclude-brown
  export route-target
   100:4
  !
 !
!
vrf vrf_cust5
 address-family ipv4 unicast
  import route-target
   100:5
  !
  export route-target
   100:5
  !
 !
!
vrf vrf_cust6
 address-family ipv6 unicast
  import route-target
   100:6
  !
  export route-policy SET_COLOR_HI_BW_exclude-brown
  export route-target
   100:6
  !
 !
!
```

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 (12.4.4.4/32) with color 20 advertised by router cafe:0:4::4.

```
Node1# show bgp vrf vrf_cust1 unicast

BGP VRF vrf_cust1, state: Active
BGP Route Distinguisher: 100:1
VRF ID: 0x60000002
BGP router identifier 1.1.1.1, local AS number 100
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000011   RD version: 100
BGP main routing table version 112
BGP NSR Initial initsync version 21 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf vrf_cust1)
*> 12.1.1.1/32        0.0.0.0                  0         32768 ?
*>i12.4.4.4/32        cafe:0:4::4 C:20         0    100      0 ?
*>i12.5.5.5/32        cafe:0:5::5 C:20         0    100      0 ?

Processed 3 prefixes, 3 paths
```

The following output displays the details for prefix 12.4.4.4/32. Note the presence of BGP extended color community 20, and that the prefix is associated with an SR policy with color 20 and BSID value of cafe:0:1:e011::.

```
Node1# show bgp vrf vrf_cust1 ipv4 unicast 12.4.4.4/32

BGP routing table entry for 12.4.4.4/32, Route Distinguisher: 100:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               100         100
Last Modified: Nov  2 15:38:08.744 for 01:30:05
Paths: (2 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    cafe:0:4::4 C:20 (bsid: cafe:0:1:e011::) (metric 30) from cafe:0:4::4 (1.1.1.4)
      Received Label 0xe0020
      Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best,
import-candidate, imported
      Received Path ID 0, Local Path ID 1, version 49
      Extended community: Color:20 RT:1:1 RT:100:1
      SR policy color 20, up, registered, bsid cafe:0:1:e011::, if-handle 0x00000000

      PSID-Type:L3, SubTLV Count:1
       SubTLV:
        T:1(Sid information), Sid:cafe:0:4::, Behavior:63, SS-TLV Count:1
         SubSubTLV:
          T:1(Sid structure):
      Source AFI: VPNv4 Unicast, Source VRF: vrf_cust1, Source Route Distinguisher: 100:1
  Path #2: Received by speaker 0
  Not advertised to any peer
  Local, (received-only)
    cafe:0:4::4 C:20 (bsid: cafe:0:1:e011::) (metric 30) from cafe:0:4::4 (1.1.1.4)
```

```
      Received Label 0xe0020
      Origin incomplete, metric 0, localpref 100, valid, internal, import-candidate,
not-in-vrf
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: Color:20 RT:1:1 RT:100:1
      SR policy color 20, up, registered, bsid cafe:0:1:e011::, if-handle 0x00000000

      PSID-Type:L3, SubTLV Count:1
       SubTLV:
        T:1(Sid information), Sid:cafe:0:4::, Behavior:63, SS-TLV Count:1
         SubSubTLV:
          T:1(Sid structure):
```

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 12.4.4.4/32 points to the BSID label corresponding to an SR policy.

```
Node1# show cef vrf vrf_cust1

Tue Nov  2 17:08:32.439 UTC

Prefix              Next Hop            Interface
------------------- ------------------- ------------------
0.0.0.0/0           drop                default handler
0.0.0.0/32          broadcast
12.1.1.1/32         receive             Loopback100
12.4.4.4/32         cafe:0:1:e011:: (via-srv6-sid) <recursive>
12.5.5.5/32         cafe:0:1:e00c:: (via-srv6-sid) <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 12.4.4.4/32. Note that the prefix is associated with an SR policy with BSID value of cafe:0:1:e011::.

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

12.4.4.4/32, version 14, SRv6 Headend, internal 0x5000001 0x30 (ptr 0x791e024c) [1], 0x0
(0x0), 0x0 (0x88f0c720)
 Updated Nov  2 15:38:08.611
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via local-srv6-sid cafe:0:1:e011::, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x78edbaf4 0x0]
    recursion-via-/64
    next hop VRF - 'default', table - 0xe0800000
    next hop cafe:0:1:e011:: via cafe:0:1:e011::/64
    SRv6 H.Encaps.Red SID-list {cafe:0:4:e002::}
```

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 20 advertised by node cafe:0:4::4.

```
Node1# show segment-routing traffic-eng policy color 20 tabular

 Color                              Endpoint  Admin   Oper                Binding
                                              State  State                   SID
```

```
------ ------------------------------------ ------ ------ ----------------------
   20                                 cafe:0:4::4     up     up         cafe:0:1:e011::
```

The following outputs show the details of the on-demand SR policy for color 20.

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.

```
Node1# show segment-routing traffic-eng policy color 20

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

Color: 20, End-point: cafe:0:4::4
  Name: srte_c_20_ep_cafe:0:4::4
  Status:
    Admin: up  Operational: up for 01:20:52 (since Nov  2 15:38:08.597)
  Candidate-paths:
    Preference: 200 (BGP ODN) (shutdown)
      Requested BSID: dynamic
      PCC info:
        Symbolic name: bgp_c_20_ep_cafe:0:4::4_discr_200
        PLSP-ID: 16
        Protection Type: unprotected-preferred
        Maximum SID Depth: 13
      Dynamic (invalid)
        Metric Type: TE,   Path Accumulated Metric: 0
      SRv6 Information:
        Locator: Node1
        Binding SID requested: Dynamic
        Binding SID behavior: End.B6.Insert.Red
    Preference: 100 (BGP ODN) (active)
      Requested BSID: dynamic
      PCC info:
        Symbolic name: bgp_c_20_ep_cafe:0:4::4_discr_100
        PLSP-ID: 15
        Protection Type: unprotected-preferred
        Maximum SID Depth: 13
      Dynamic (pce cafe:0:2::2) (valid)
        Metric Type: TE,   Path Accumulated Metric: 12
          SID[0]: Node: cafe:0:2:: Behavior: uN (PSP/USD) (48)
                  LBL:32 LNL:16 FL:0 AL:0
                  Address cafe:0:2::2
          SID[1]: Node: cafe:0:4:: Behavior: uN (PSP/USD) (48)
                  LBL:32 LNL:16 FL:0 AL:0
                  Address cafe:0:4::4
      SRv6 Information:
        Locator: Node1
        Binding SID requested: Dynamic
        Binding SID behavior: End.B6.Insert.Red
  Attributes:
    Binding SID: cafe:0:1:e011::
    Forward Class: Not Configured
    Steering labeled-services disabled: no
    Steering BGP disabled: no
    IPv6 caps enable: yes
    Invalidation drop enabled: no
```

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 20 advertised by node cafe:0:4::4.

```
Node1# show segment-routing traffic-eng forwarding policy color 20 tabular

SR-TE Policy Forwarding database
--------------------------------

Color: 20, End-point: cafe:0:4::4
  Name: srte_c_20_ep_cafe:0:4::4
  Binding SID: cafe:0:1:e011::
  Active LSP:
    Candidate path:
      Preference: 100 (BGP ODN)
    Segment lists:
      SL[0]:
        Name: dynamic
        Switched Packets/Bytes: ?/?
        Paths:
          Path[0]:
            Out-SID                                    Out-Interface   Next Hop
                     Bytes Switched
            ---------------------------------------- ---------------
----------------------------------- --------------------
            cafe:0:2::                                Hu0/0/0/0       fe80::e00:ff:fece:2200
                    ?
          Path[1]:
            Out-SID                                    Out-Interface   Next Hop
                     Bytes Switched
            ---------------------------------------- ---------------
----------------------------------- --------------------
            cafe:0:4::                                Hu0/0/0/1       fe80::1600:ff:fef4:8300
                    ?                    (!)

  Policy Packets/Bytes Switched: ?/?
  (!): FRR pure backup


Node1# show segment-routing traffic-eng forwarding policy color 20 detail

SR-TE Policy Forwarding database
--------------------------------

Color: 20, End-point: cafe:0:4::4
  Name: srte_c_20_ep_cafe:0:4::4
  Binding SID: cafe:0:1:e011::
  Active LSP:
    Candidate path:
      Preference: 100 (BGP ODN)
    Segment lists:
      SL[0]:
        Name: dynamic
        Switched Packets/Bytes: ?/?
        Paths:
          Path[0]:
            Outgoing Interfaces: HundredGigE0/0/0/0
            Next Hop: fe80::e00:ff:fece:2200
            FRR Pure Backup: No
            ECMP/LFA Backup: No
```

```
    SID stack (Top -> Bottom): {cafe:0:2::}
    Path-id: 1 (Protected), Backup-path-id: 65, Weight: 0
  Path[1]:
    Outgoing Interfaces: HundredGigE0/0/0/1
    Next Hop: fe80::1600:ff:fef4:8300
    FRR Pure Backup: Yes
    ECMP/LFA Backup: Yes
    SID stack (Top -> Bottom): {cafe:0:4::, cafe:0:2::}
    Path-id: 65 (Pure-Backup), Weight: 0

Policy Packets/Bytes Switched: ?/?
```

Manually Provisioned SRv6 Policy

Manually provisioned SRv6 policies are configured on the head-end router. These policies can use dynamic paths or explicit paths. See the SRv6-TE Policy Path Types, on page 18 section for information on manually provisioning an SRv6 policy using dynamic or explicit paths.

SRv6-TE Policy Path Types

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

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 SRv6-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 (uB6) – A uB6 SID conflict occurs when there are different SRv6 policies with the same uB6 SID. In this case, the policy that is installed first gets the uB6 SID 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.

The following SRv6-TE policy path types are supported in this release:

- Dynamic Paths, on page 19

Dynamic Paths

Behaviors and Limitations

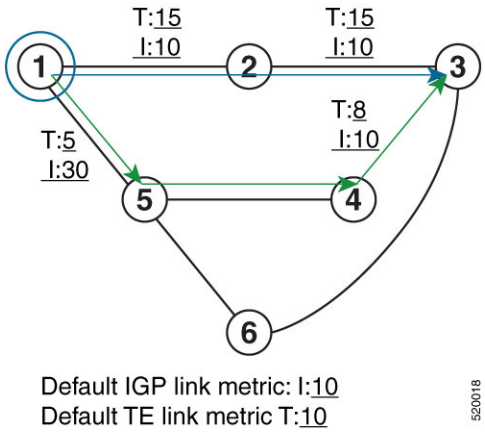
For a dynamic path that traverses a specific interface between nodes (segment), the algorithm may encode this segment using an Adjacency uSID (uA SID).

Optimization Objectives

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

- Hopcount — Use the least number of hops for path computation.
- IGP metric — Refer to the "Implementing IS-IS" and "Implementing OSPF" chapters in the *Routing Configuration Guide for Series Routers*.
- TE metric — See the Configure Interface TE Metrics section for information about configuring TE metrics.
- Delay (latency) — See the Configure Performance Measurement chapter for information about measuring delay for links or SRv6 policies.

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



- The blue path uses the minimum IGP metric: Min-Metric (1 → 3, IGP) = uSID-list {cafe:0:**3**::}; cumulative IGP metric: 20
- The green path uses the minimum TE metric: Min-Metric (1 → 3, TE) = uSID-list {cafe:0:**5**:**4**:**3**::}; 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:

Note For path-computation on PCE, configuring both affinity and disjoint-path is not supported.

- 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 SRv6 policy path and link colors. SRv6-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 section for information on named interface link admin groups and SRv6-TE Affinity Maps.
- Disjoint — SRv6-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).
- Segment protection-type behavior — You can control whether protected or unprotected segments are used when encoding the SID-list of an SRv6 policy candidate path. The types of segments that could be used when building a SID-list include uSIDs and adjacency SIDs (uA SID).

Named Interface Link Admin Groups and SRv6-TE Affinity Maps

Named Interface Link Admin Groups and SRv6-TE Affinity Maps provide a simplified and more flexible means of configuring link attributes and path affinities to compute paths for SRv6-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 SRv6-TE Affinity Maps let you assign, or map, up to 256 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 SRv6-TE Affinity Maps

Use the **segment-routing traffic-eng interface** *interface* **affinity name** *NAME* command to assign affinity to interfaces. Configure this on routers with interfaces that have an associated admin group attribute.

Use the **segment-routing traffic-eng affinity-map name** *NAME* **bit-position** *bit-position* command 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 SRv6-TE head-ends for SR policies that include affinity constraints.

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 (SRv6-TE head-end or transit node) with colored interfaces.

```
Router(config)# segment-routing
Router(config-sr)# traffic-eng
Router(config-sr-te)# interface HundredGigE0/0/0/0
Router(config-sr-if)# affinity
Router(config-sr-if-affinity)# name brown
Router(config-sr-if-affinity)# exit
Router(config-sr-if)# exit

Router(config-sr-te)# affinity-map
Router(config-sr-te-affinity-map)# name brown bit-position 1
```

Running Config

```
segment-routing
 traffic-eng
  interface HundredGigE0/0/0/0
   affinity
    name brown
   !
  !
  affinity-map
   name brown bit-position 1
  !
end
```

Configure SRv6 Policy with Dynamic Path

To configure a SRv6-TE policy with a dynamic path, optimization objectives, and affinity constraints, complete the following configurations:

1. Define the optimization objectives. See the Optimization Objectives section.

2. Define the constraints. See the Constraints section.
3. Create the policy.

Behaviors and Limitations

For a dynamic path that traverses a specific interface between nodes (segment), the algorithm may encode this segment using an Adjacency micro-SID (uA SID).

Configuration

Create the SRv6-TE Policy

- Use the **segment-routing traffic-eng policy** *policy* **srv6 color** *color* **end-point ipv6** *ipv6-address* command to configure the SRv6-TE policy color and IPv6 end-point address.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te srv6 color 50
end-point ipv6 cafe:0:4::4
```

Specify Customized Locator and Source Address

- (Optional) Use the **segment-routing traffic-eng policy** *policy* **srv6 locator** *locator* **binding-sid dynamic behavior ub6-insert-reduced** command to specify the customized per-policy locator and BSID behavior. If you do not specify a customized per-policy locator and BSID behavior, the policy will use the global locator and BSID behavior.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te srv6 locator node1
 binding-sid dynamic behavior ub6-insert-reduced
```

- (Optional) Use the **segment-routing traffic-eng policy** *policy* **source-address ipv6** *ipv6-address* command to specify the customized IPv6 source address for candidate paths. If you do not specify a customized per-policy source address, the policy will use the local IPv6 source address.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te srv6 source-address
 ipv6 cafe:0:1::1
```

Enable Dynamic Path Computed by the SR-PCE

- Use the **segment-routing traffic-eng policy** *policy* **candidate-paths preference** *preference* **dynamic pcep** command to specify the candidate path preference and enable dynamic path computed by the SR-PCE. The range for *preference* is from 1 to 65535.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te candidate-paths
preference 100 dynamic pcep
```

Configure Dynamic Path Optimization Objectives

- Use the **segment-routing traffic-eng policy** *policy* **candidate-paths preference** *preference* **dynamic metric type** {**igp** | **te** | **latency**} command to configure the metric for use in path computation.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te candidate-paths
preference 100 metric type te
```

- Use the **segment-routing traffic-eng policy** *policy* **candidate-paths preference** *preference* **dynamic metric margin** {**absolute** *value*| **relative** *percent*} command to configure the dynamic path metric margin. The range for *value* and *percent* is from 0 to 2147483647.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te candidate-paths
preference 100 metric margin absolute 5
```

Configure Dynamic Path Constraints

Note Disjoint-path and affinity constraints cannot be configured at the same time.

- Use the **segment-routing traffic-eng policy** *policy* **candidate-paths preference** *preference* **constraints affinity** {**exclude-any** | **include-all** | **include-any**} *name* command to configure the affinity constraints.

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te candidate-paths
preference 100 constraints affinity exclude-any name brown
```

- Use the **segment-routing traffic-eng policy** *policy* **candidate-paths preference** *preference* **constraints 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

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te candidate-paths
preference 100 constraints disjoint-path group-id 775 type link
```

- Use the **segment-routing traffic-eng policy** *policy* **candidate-paths preference** *preference* **constraints segments protection** {**protected-only** | **protected-preferred** | **unprotected-only** | **unprotected-preferred**} command to configure the segment protection-type behavior.

  The following example shows how to configure the policy with a SID-list that must be encoded using protected segments:

```
Node1(config)# segment-routing traffic-eng policy pol_node1_node4_te candidate-paths
preference 100 constraints segments protection protected-only
```

Examples

The following example shows a configuration of an SRv6 policy at an SRv6-TE head-end router using the global locator and source address. The policy has a dynamic path with optimization objectives and affinity constraints computed by the SR-PCE.

```
Node1(config)# segment-routing
Node1(config-sr)# traffic-eng
Node1(config-sr-te)# srv6
Node1(config-sr-te-srv6)# locator Node1 binding-sid dynamic behavior ub6-insert-reduced
Node1(config-sr-te-srv6)# exit
Node1(config-sr-te)# candidate-paths
Node1(config-sr-te-candidate-path)# all
Node1(config-sr-te-candidate-path-type)# source-address ipv6 cafe:0:1::1
Node1(config-sr-te-candidate-path-type)# exit
Node1(config-sr-te-candidate-path)# exit

Node1(config-sr-te)# policy pol_node1_node4_te
Node1(config-sr-te-policy)# color 20 end-point ipv6 cafe:0:4::4
Node1(config-sr-te-policy)# candidate-paths
Node1(config-sr-te-policy-path)# preference 100
```

```
Node1(config-sr-te-policy-path-pref)# dynamic
Node1(config-sr-te-pp-info)# pcep
Node1(config-sr-te-path-pcep)# exit
Node1(config-sr-te-pp-info)# metric type te
Node1(config-sr-te-pp-info)# exit
Node1(config-sr-te-policy-path-pref)# constraints
Node1(config-sr-te-path-pref-const)# affinity
Node1(config-sr-te-path-pref-const-aff)# exclude-any
Node1(config-sr-te-path-pref-const-aff-rule)# name brown
```

Running Config

```
segment-routing
 traffic-eng
  srv6
   locator Node1 binding-sid dynamic behavior ub6-insert-reduced
  !
  candidate-paths
   all
    source-address ipv6 cafe:0:1::1
   !
  !
  policy pol_node1_node4_te
   color 20 end-point ipv6 cafe:0:4::4
   candidate-paths
    preference 100
     dynamic
      pcep
      !
      metric
       type te
      !
     !
     constraints
      affinity
       exclude-any
        name brown
       !
      !
     !
    !
   !
  !
 !
!
```

The following example shows a configuration of a manual SRv6 policy at an SRv6-TE head-end router with customized locator and source address. The policy has a dynamic path with optimization objectives and affinity constraints computed by the SR-PCE.

```
Node1(config)# segment-routing
Node1(config-sr)# traffic-eng
Node1(config-sr-te)# policy pol_node1_node4_te
Node1(config-sr-te-policy)# source-address ipv6 cafe:0:1::1
Node1(config-sr-te-policy)# srv6
Node1(config-sr-te-policy-srv6)# locator Node1 binding-sid dynamic behavior ub6-insert-reduced
Node1(config-sr-te-policy-srv6)# exit
Node1(config-sr-te-policy)# color 20 end-point ipv6 cafe:0:4::4
Node1(config-sr-te-policy)# candidate-paths
Node1(config-sr-te-policy-path)# preference 100
Node1(config-sr-te-policy-path-pref)# dynamic
Node1(config-sr-te-pp-info)# pcep
Node1(config-sr-te-path-pcep)# exit
```

```
Node1(config-sr-te-pp-info)# metric type te
Node1(config-sr-te-pp-info)# exit
Node1(config-sr-te-policy-path-pref)# constraints
Node1(config-sr-te-path-pref-const)# affinity
Node1(config-sr-te-path-pref-const-aff)# exclude-any
Node1(config-sr-te-path-pref-const-aff-rule)# name brown
```

Running Config

```
segment-routing
 traffic-eng
  policy pol_node1_node4_te
   srv6
    locator Node1 binding-sid dynamic behavior ub6-insert-reduced
   !
   source-address ipv6 cafe:0:1::1
   color 20 end-point ipv6 cafe:0:4::4
   candidate-paths
    preference 100
     dynamic
      pcep
      !
      metric
       type te
      !
     !
     constraints
      affinity
       exclude-any
        name brown
       !
      !
     !
    !
   !
  !
 !
!
```

Verification

```
Node1# show segment-routing traffic-eng policy color 20

SR-TE policy database
---------------------
Color: 20, End-point: cafe:0:4::4
  Name: srte_c_20_ep_cafe:0:4::4
  Status:
    Admin: up  Operational: down for 00:00:09 (since Nov  5 20:10:26.158)
  Candidate-paths:
    Preference: 100 (configuration)
      Name: pol_node1_node4_te
      Requested BSID: dynamic
      PCC info:
        Symbolic name: cfg_pol_node1_node4_te_discr_100
        PLSP-ID: 1
        Protection Type: unprotected-preferred
        Maximum SID Depth: 13
      Dynamic (pce cafe:0:2::2) (valid)
        Metric Type: NONE,   Path Accumulated Metric: 0
      SRv6 Information:
        Locator: Node1
        Binding SID requested: Dynamic
```

```
      Binding SID behavior: End.B6.Insert.Red
Attributes:
  Forward Class: 0
  Steering labeled-services disabled: no
  Steering BGP disabled: no
  IPv6 caps enable: yes
  Invalidation drop enabled: no
```

Protocols

Path Computation Element Protocol

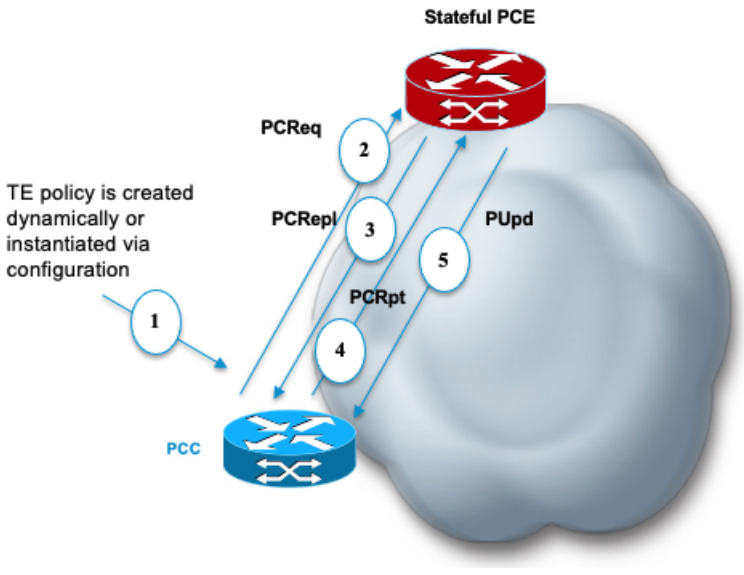
The path computation element protocol (PCEP) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end label switched paths (LSPs) sourced from the PCC to a PCE peer. The PCE can request the PCC to update and modify parameters of LSPs it controls.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

A PCEP channel is established over TCP and has its own light-weight Keep-Alive (KA) mechanism

Upon configuring a PCE peer, a PCC opens a PCEP session to the PCE, and the PCE accepts the PCEP sessions if the following conditions are satisfied:

- The total number of PCEP sessions does not exceed the limit on the total number of PCEP sessions on the PCE.
- The KA interval indicated by PCC is acceptable to the PCE.

Sample Workflow with Stateful PCEP



1. The PCC is configured to instantiate an SRv6-TE policy.

2. The PCC sends a PCEP Path Computation Request (PCReq) to the PCE, requesting a path by specifying path attributes, optimization objectives, and constraints.
3. The PCE stores the request, computes a TE metric shortest-path, and returns the computed SID list in a PCEP Path Computation Reply (PCRepl).
4. The PCC allocates a BSID and activates the SR Policy using the SID list computed by the PCE. The PCC sends a Path Computation Report (PCRpt) to the PCE, delegating the SR Policy to the PCE and including BSID.
5. The PCE updates the paths when required (for example, following a multi-domain topology change that impacts connectivity).

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 ipv6 ipv6-local-source-address
Router(config-sr-te-pcc)# pce address ipv6 ipv6-PCE-address[precedence value]
```

Configure PCEP Authentication

TCP Message Digest 5 (MD5) authentication has been used for authenticating PCEP (TCP) sessions by using a clear text or encrypted password. This feature introduces support for TCP Authentication Option (TCP-AO), which replaces the TCP MD5 option.

TCP-AO uses Message Authentication Codes (MACs), which provides the following:

- Protection against replays for long-lived TCP connections
- More details on the security association with TCP connections than TCP MD5
- A larger set of MACs with minimal system and operational changes

TCP-AO is compatible with Master Key Tuple (MKT) configuration. TCP-AO also protects connections when using the same MKT across repeated instances of a connection. TCP-AO protects the connections by using traffic key that are derived from the MKT, and then coordinates changes between the endpoints.

Note TCP-AO and TCP MD5 are never permitted to be used simultaneously. TCP-AO supports IPv6, and is fully compatible with the proposed requirements for the replacement of TCP MD5.

TCP Message Digest 5 (MD5) Authentication

Use the **password** {**clear** | **encrypted**} *LINE* command to enable TCP MD5 authentication for all PCEP peers. Any TCP segment coming from the PCC that does not contain a MAC matching the configured password will be rejected. Specify if the password is encrypted or clear text

```
Router(config-sr-te-pcc)# pce address ipv6 ipv6-PCE-address[password {clear | encrypted}
LINE]
```

TCP Authentication Option (TCP-AO)

Use the **tcp-ao** *key-chain* [**include-tcp-options**] command to enable TCP Authentication Option (TCP-AO) authentication for all PCEP peers. Any TCP segment coming from the PCC that does not contain a MAC matching the configured key chain will be rejected. Use the **include-tcp-options** keyword to include other TCP options in the header for MAC calculation.

```
Router(config-sr-te-pcc)# pce address ipv6 ipv6-PCE-address tcp-ao key-chain
[include-tcp-options]
```

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 MSD is expressed as a number uSIDs. The number of uSID is expressed as a number of carriers and the number of uSID per carrier.

The default MSD *value* is equal to the maximum MSD supported by the platform (6 — 1 carrier, 6 uSIDs per carrier).

```
Router(config-sr-te-srv6)# 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 uSID stack size is verified against the MSD requirement.

- If the uSID 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 uSID 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 uSIDs, but a sub-optimal path exists (with accumulated metric of 110) requiring 4 uSIDs, 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 2 PCEP servers (PCE) with different precedence values. The PCE with IP address cafe:0:2::2 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.

```
Node1(config)# segment-routing
Node1(config-sr)# traffic-eng
Node1(config-sr-te)# pcc
Node1(config-sr-te-pcc)# source-address ipv6 cafe:0:1::1
Node1(config-sr-te-pcc)# pce address ipv6 cafe:0:2::2
Node1(config-pcc-pce)# precedence 10
Node1(config-pcc-pce)# exit
Node1(config-sr-te-pcc)# pce address ipv6 cafe:0:3::3
Node1(config-pcc-pce)# precedence 20
Node1(config-pcc-pce)# exit
Node1(config-sr-te-pcc)# report-all
Node1(config-sr-te-pcc)# exit
Node1(config-sr-te)# srv6
Node1(config-sr-te-srv6)# maximum-sid-depth 5
Node1(config-sr-te-srv6)# exit
Node1(config-sr-te)# logging
Node1(config-sr-te-log)# policy status
Node1(config-sr-te-log)# exit
Node1(config-sr-te)#
```

Running Config

```
segment-routing
 traffic-eng
  srv6
   maximum-sid-depth 5
  !
  logging
   policy status
  !
  pcc
   source-address ipv6 cafe:0:1::1
   pce address ipv6 cafe:0:2::2
    precedence 10
   !
   pce address ipv6 cafe:0:3::3
    precedence 20
   !
   report-all
  !
 !
!
```

Verification

```
Node1# show segment-routing traffic-eng pcc ipv6 peer brief

             Address   Precedence         State    Learned From
-------------------- ------------ ------------ ---------------
         cafe:0:2::2           10           up          config
         cafe:0:3::3           20           up          config


Node1# show segment-routing traffic-eng pcc ipv6 peer detail

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

```
Peer address: cafe:0:2::2
  Precedence: 10, (best PCE)
  State up
  Capabilities: Stateful, Update, Segment-Routing, Instantiation
  PCEP has been up for: 01:22:23
  Local keepalive timer is 30 seconds
  Remote keepalive timer is 30 seconds
  Local dead timer is 120 seconds
  Remote dead timer is 120 seconds
  Authentication: None
  Statistics:
    Open messages:      rx 1        |  tx 1
    Close messages:     rx 0        |  tx 0
    Keepalive messages: rx 164      |  tx 163
    Error messages:     rx 0        |  tx 0
    Report messages:    rx 0        |  tx 110
    Update messages:    rx 36       |  tx 0

Peer address: cafe:0:3::3
  Precedence: 20
  State up
  Capabilities: Stateful, Update, Segment-Routing, Instantiation
  PCEP has been up for: 01:21:48
  Local keepalive timer is 30 seconds
  Remote keepalive timer is 30 seconds
  Local dead timer is 120 seconds
  Remote dead timer is 120 seconds
  Authentication: None
  Statistics:
    Open messages:      rx 1        |  tx 1
    Close messages:     rx 0        |  tx 0
    Keepalive messages: rx 164      |  tx 162
    Error messages:     rx 0        |  tx 0
    Report messages:    rx 0        |  tx 82
    Update messages:    rx 0        |  tx 0
```

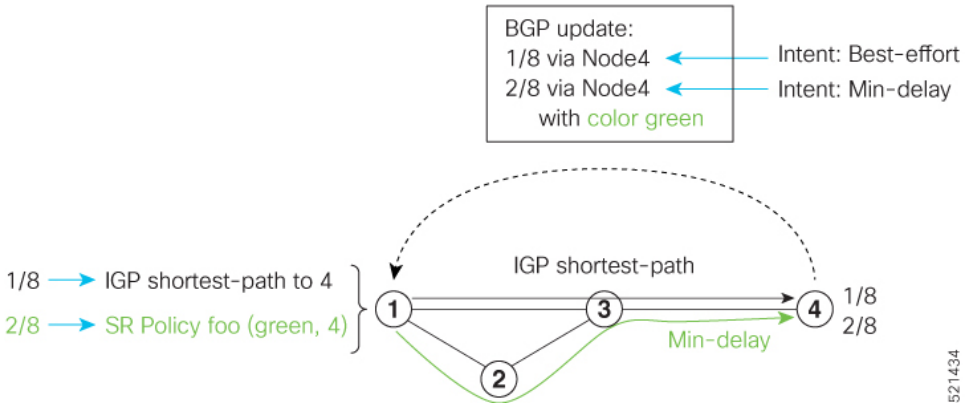
Traffic Steering

Automated Steering

Automated steering (AS) allows service traffic to be automatically steered onto the required transport SLA path programmed by an SRv6 policy.

With AS, BGP automatically steers traffic onto an SRv6 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 SRv6 policy, BGP automatically installs the route resolving onto the BSID of the matching SRv6 policy. Recall that an SRv6 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 SRv6 policy, the BGP process installs the route on the SRv6 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