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.

> **Note** Configuring SR-TE policies with 3 or more labels and an L2 Transport Interface on the same network processing unit (NPU) can cause traffic loss.

About SR-TE Policies

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 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.

There are two types of SR-TE policies: dynamic and explicit.

Local Dynamic SR-TE Policy

When you configure local dynamic SR-TE, the head-end locally calculates the path to the destination address. Dynamic path calculation results in a list of interface IP addresses that traffic engineering (TE) maps to adj-SID labels. Routes are learned by way of forwarding adjacencies over the TE tunnel.

Explicit SR-TE Policy

An explicit path is a list of IP addresses or labels, each representing a node or link in the explicit path. This feature is enabled through the **explicit-path** command that allows you to create an explicit path and enter a configuration submode for specifying the path.

How to Configure SR-TE Policies

This section contains the following procedures:

• Configure Local Dynamic SR-TE Policy, on page 2
• Configure Explicit SR-TE Policy, on page 3

Configure Local Dynamic SR-TE Policy

This task explains how to configure a local dynamic SR-TE policy.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*
4. **destination** *ip-address*
5. **path-option** *preference-priority* **dynamic segment-routing**
6. **path-protection**
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**interface tunnel-te** *tunnel-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te22`	Configures the tunnel interface.
Step 3	**ipv4 unnumbered** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0`	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 4	**destination** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination`	Assigns a destination address on the new tunnel.

	Command or Action	Purpose
	`192.168.0.2`	
Step 5	**path-option** *preference-priority* **dynamic segment-routing** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing`	Sets the path option to dynamic and assigns the path ID.
Step 6	**path-protection** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-protection`	Enables path protection on the tunnel-te interface.
Step 7	**commit**	

This completes the configuration of the dynamic SR-TE policy.

Configure Explicit SR-TE Policy

This task explains how to configure an explicit SR-TE policy.

SUMMARY STEPS

1. **configure**
2. **explicit-path name** *path-name*
3. **index** *index* {**next-address** *ip-address* | **next-label** *label*}
4. **exit**
5. **interface tunnel-te** *tunnel-id*
6. **ipv4 unnumbered** *type interface-path-id*
7. **destination** *ip-address* [**verbatim**]
8. **path-option** *preference-priority* **explicit name** *path-name* **segment-routing**
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**explicit-path name** *path-name* **Example:**	Enters a name for the explicit path and enters the explicit path configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# explicit-path name r1r6_exp`	

Step 3	**index** *index* { **next-address** *ip-address* | **next-label** *label* }
	Example:
	`RP/0/RP0/CPU0:router(config-expl-path)# index 1 next-label 16001`
	`RP/0/RP0/CPU0:router(config-expl-path)# index 2 next-label 16006`
		Specifies a label or an address in an explicit path of a tunnel.
		Note • You can include multiple addresses, labels, or both. However, you cannot configure addresses after you have configured labels. Once you start configuring labels, you need to continue with labels.
		• Each entry must have a unique index.
		• If the first hop is specified as next-label, that label must be an Adj-SID of the head-end or a prefix-SID label value known by the head-end.

Step 4	**exit**	
Step 5	**interface tunnel-te** *tunnel-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te22`	Configures the tunnel interface.
Step 6	**ipv4 unnumbered** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0`	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 7	**destination** *ip-address* [**verbatim**] **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2`	Assigns a destination address on the new tunnel. Typically, the tunnel destination must have a match in the routing information base (RIB). For inter-area or inter-domain policies to destinations that are otherwise not reachable, use the **verbatim** option to disable the RIB verification on a tunnel destination.
Step 8	**path-option** *preference-priority* **explicit name** *path-name* **segment-routing** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name r1r6_exp segment-routing`	Specifies the explicit path name and assigns the path ID.
Step 9	**commit**	

This completes the configuration of the explicit SR-TE policy.

Steering Traffic into an SR-TE Policy

This section describes the following traffic steering methods:

Static Routes

Static routes can use the segment routing tunnel as a next-hop interface. Both IPv4 and IPv6 prefixes can be routed through the tunnel.

A static route to a destination with a prefix-SID removes the IGP-installed SR-forwarding entry of that prefix.

Autoroute Announce

The SR-TE policy can be advertised into an IGP as a next hop by configuring the autoroute announce statement on the source router. The IGP then installs routes in the Routing Information Base (RIB) for shortest paths that involve the tunnel destination. Autoroute announcement of IPv4 prefixes can be carried through either OSPF or IS-IS. Autoroute announcement of IPv6 prefixes can be carried only through IS-IS.

Autoroute Destination

Autoroute destination allows you to automatically route traffic through a segment routing tunnel instead of manually configuring static routes. Multiple autoroute destination addresses can be added in the routing information base (RIB) per tunnel.

Static routes are always added with zero cost metric, which can result in traffic that is mapped on multiple tunnels to always load-balance due to ECMP. This load-balancing may be undesirable when some of those tunnels have sub-optimal paths. With autoroute destination, only the tunnel whose IGP cost to its endpoint is lowest will be considered for carrying traffic.

- **Interaction Between Static Routes and Autoroute Destination**

  If there is a manually configured static route to the same destination as a tunnel with autoroute destination enabled, traffic for that destination is load-shared between the static route and the tunnel with autoroute destination enabled.

- **Interaction Between Autoroute Announce and Autoroute Destination**

  For intra-area tunnels, if a tunnel is configured with both autoroute announce and autoroute destination, the tunnel is announced to the RIB by both the IGP and the static process. RIBs prefer static routes, not IGP routes, so the autoroute destination features takes precedence over autoroute announce.

Configure Static Routes

This task explains how to configure a static route.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*

4. **destination** *ip-address*
5. **path-option** *preference-priority* **dynamic segment-routing**
6. **exit**
7. **router static**
8. **address-family ipv4 unicast**
9. *prefix mask interface-type interface-instance*
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**interface tunnel-te** *tunnel-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te22`	Configures the tunnel interface.
Step 3	**ipv4 unnumbered** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0`	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 4	**destination** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2`	Assigns a destination address on the new tunnel.
Step 5	**path-option** *preference-priority* **dynamic segment-routing** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing`	Sets the path option to dynamic and assigns the path ID.
Step 6	**exit**	
Step 7	**router static** **Example:**	Configures the static route and enters static configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# router static`	
Step 8	**address-family ipv4 unicast** **Example:** `RP/0/RP0/CPU0:router(config-static)# address-family ipv4 unicast`	Enters address family mode.
Step 9	*prefix mask interface-type interface-instance* **Example:** `RP/0/RP0/CPU0:router(config-static-af)# 192.168.0.2/32 tunnel-te22`	Specifies the destination prefix is directly reachable through the tunnel interface.
Step 10	**commit**	

This completes the configuration of the static route.

Configure Autoroute Announce

This task explains how to configure autoroute announce to steer traffic through the SR-TE policy.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*
4. **autoroute announce**
5. **destination** *ip-address*
6. **path-option** *preference-priority* **dynamic segment-routing**
7. **path-protection**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**interface tunnel-te** *tunnel-id* **Example:**	Configures the tunnel interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# interface tunnel-te22`	
Step 3	**ipv4 unnumbered** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0`	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 4	**autoroute announce** **Example:** `RP/0/RP0/CPU0:router(config-if)# autoroute announce`	Enables messages that notify the neighbor nodes about the routes that are forwarding.
Step 5	**destination** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2`	Assigns a destination address on the new tunnel.
Step 6	**path-option** *preference-priority* **dynamic segment-routing** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic segment-routing`	Sets the path option to dynamic and assigns the path ID.
Step 7	**path-protection** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-protection`	Enables path protection on the tunnel-te interface.
Step 8	**commit**	

Configure Autoroute Destination

This task explains how to configure autoroute destination to steer traffic through the SR-TE policy.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*

4. **autoroute destination** *destination-ip-address*
5. **destination** *ip-address*
6. **path-option** *preference-priority* **dynamic segment-routing**
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**interface tunnel-te** *tunnel-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te22`	Configures the tunnel interface.
Step 3	**ipv4 unnumbered** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0`	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.

Step 4	**autoroute destination** *destination-ip-address*
	Example:
	`RP/0/RP0/CPU0:router(config-if)# autoroute destination 192.168.0.1`
	`RP/0/RP0/CPU0:router(config-if)# autoroute destination 192.168.0.2 (the default route)`
	`RP/0/RP0/CPU0:router(config-if)# autoroute destination 192.168.0.3`
	`RP/0/RP0/CPU0:router(config-if)# autoroute destination 192.168.0.4`
		(Optional) Adds a route (destination-ip-address) in the RIB with the tunnel as outgoing interface to the tunnel destination.

Step 5	**destination** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.0.2`	Assigns a destination address on the new tunnel.
Step 6	**path-option** *preference-priority* **dynamic segment-routing** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1`	Sets the path option to dynamic and assigns the path ID.

	Command or Action	Purpose
	`dynamic segment-routing`	
Step 7	**commit**	

BGP SR-TE

SR-TE can be used by data center (DC) operators to provide different levels of Service Level Assurance (SLA). Setting up SR-TE paths using BGP (BGP SR-TE) simplifies DC network operation without introducing a new protocol for this purpose.

Explicit BGP SR-TE

Explicit BGP SR-TE uses an SR-TE policy (identified by a unique color ID) that contains a list of explicit paths with SIDs that correspond to each explicit path. A BGP speaker signals an explicit SR-TE policy to a remote peer, which triggers the setup of a TE tunnel with specific characteristics and explicit paths. On the receiver side, a TE tunnel that corresponds to the explicit path is setup by BGP. The packets for the destination mentioned in the BGP update follow the explicit path described by the policy. Each policy can include multiple explicit paths, and TE will create a tunnel for each path.

Note For more information on routing policies and routing policy language (RPL), refer to the "Implementing Routing Policy" chapter in the *Routing Configuration Guide for Cisco NCS 5500 Series Routers*.

IPv4 and IPv6 SR policies 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 Explicit BGP SR-TE

Perform this task to configure explicit BGP SR-TE:

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
Step 1	**configure**	
Step 2	**router bgp** *as-number* **Example:** `RP/0/RSP0/CPU0:router(config)# router bgp 65000`	Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 3	**bgp router-id** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1`	Configures the local router with a specified router ID.
Step 4	**address-family** {**ipv4** | **ipv6**} **sr-policy** **Example:** `RP/0/RSP0/CPU0:router(config-bgp)# address-family ipv4 sr-policy`	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode.
Step 5	**exit**	
Step 6	**neighbor** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.10.0.1`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 7	**remote-as** *as-number* **Example:** `RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 1`	Creates a neighbor and assigns a remote autonomous system number to it.
Step 8	**address-family** {**ipv4** | **ipv6**} **sr-policy** **Example:** `RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family ipv4 sr-policy`	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode.
Step 9	**route-policy** *route-policy-name* {**in** | **out**} **Example:**	Applies the specified policy to IPv4 or IPv6 unicast routes.

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# **route-policy pass out**	

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 1.1.1.1
 !
 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 1.1.1.1
 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
  !
 !
!
```

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.

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.

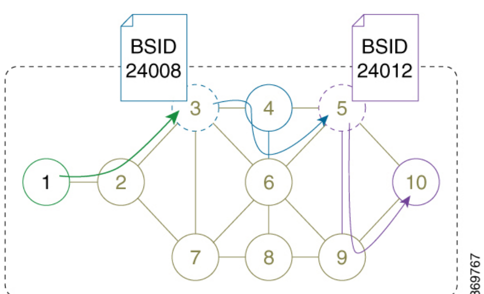
BSID can be used in the following cases:

- 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.

Stitching SR-TE Polices Using Binding SID: Example

In this intra-domain example, three SR-TE policies are stitched together to form a seamless end-to-end path from node 1 to node 10.

Figure 1: Intra-Domain Topology



Step 1 Configure an SR-TE policy on node 5 to node 10 via node 9. Node 5 automatically allocates a binding-SID (24012) for the SR-TE policy.

Example:

```
RP/0/0/CPU0:xrvr-5(config)# explicit-path name PATH5-9_10
RP/0/0/CPU0:xrvr-5(config-expl-path)# index 10 next-address strict ipv4 unicast 192.168.59.9
RP/0/0/CPU0:xrvr-5(config-expl-path)# index 20 next-address strict ipv4 unicast 10.1.1.10
RP/0/0/CPU0:xrvr-5(config-expl-path)# exit

RP/0/0/CPU0:xrvr-5(config)# interface tunnel-te1
RP/0/0/CPU0:xrvr-5(config-if)# ipv4 unnumbered Loopback0
RP/0/0/CPU0:xrvr-5(config-if)# destination 10.1.1.10
RP/0/0/CPU0:xrvr-5(config-if)# path-option 1 explicit name PATH5-9_10 segment-routing
```

```
RP/0/0/CPU0:xrvr-5(config-if)# commit

RP/0/0/CPU0:xrvr-5# show mpls traffic-eng tunnels 1 detail
Name: tunnel-te1  Destination: 10.1.1.10  Ifhandle:0x680
  Signalled-Name: xrvr-5_t1
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1, (Segment-Routing) type dynamic (Basis for Setup, path weight 10)
<...>
 Binding SID: 24012
<...>
    Segment-Routing Path Info (IS-IS 1 level-2)
      Segment0[Link]: 192.168.59.5 - 192.168.59.9, Label: 24007
      Segment1[Node]: 10.1.1.10, Label: 16010
```

Step 2 Configure an SR-TE policy on node 3 to node 5 via node 4 and Link4-6, and push the binding-SID of the SR-TE policy at node 5 (24012) to stitch to the SR-TE policy on node 5. Node 3 automatically allocates a binding-SID (24008) for this SR-TE policy.

Example:

```
RP/0/0/CPU0:xrvr-3(config)# explicit-path name PATH4_4-6_5_BSID
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 10 next-address strict ipv4 unicast 10.1.1.4
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 20 next-address strict ipv4 unicast 192.168.46.6
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 30 next-address strict ipv4 unicast 10.1.1.5
RP/0/0/CPU0:xrvr-3(config-expl-path)# index 40 next-label 24012
RP/0/0/CPU0:xrvr-3(config-expl-path)# exit

RP/0/0/CPU0:xrvr-3(config)# interface tunnel-te1
RP/0/0/CPU0:xrvr-3(config-if)# ipv4 unnumbered Loopback0
RP/0/0/CPU0:xrvr-3(config-if)# destination 10.1.1.10
RP/0/0/CPU0:xrvr-3(config-if)# path-option 1 explicit name PATH4_4-6_5_BSID segment-routing
RP/0/0/CPU0:xrvr-3(config-if)# commit

RP/0/0/CPU0:xrvr-3# show mpls traffic-eng tunnels 1 detail
Name: tunnel-te1  Destination: 10.1.1.10  Ifhandle:0x780
  Signalled-Name: xrvr-3_t1
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1, (Segment-Routing) type explicit PATH4_6_5 (Basis for Setup)
<...>
 Binding SID: 24008
<...>
    Segment-Routing Path Info (IS-IS 1 level-2)
      Segment0[Node]: 10.1.1.4, Label: 16004
      Segment1[Link]: 192.168.46.4 - 192.168.46.6, Label: 24003
      Segment2[Node]: 10.1.1.5, Label: 16005
      Segment3[ - ]: Label: 24012
```

Step 3 Configure an SR-TE policy on node 1 to node 3 and push the binding-SID of the SR-TE policy at node 3 (24008) to stitch to the SR-TE policy on node 3.

Example:

```
RP/0/0/CPU0:xrvr-1(config)# explicit-path name PATH3_BSID
RP/0/0/CPU0:xrvr-1(config-expl-path)# index 10 next-address strict ipv4 unicast 10.1.1.3
RP/0/0/CPU0:xrvr-1(config-expl-path)# index 20 next-label 24008
RP/0/0/CPU0:xrvr-1(config-expl-path)# exit

RP/0/0/CPU0:xrvr-1(config)# interface tunnel-te1
```

```
RP/0/0/CPU0:xrvr-1(config-if)# ipv4 unnumbered Loopback0
RP/0/0/CPU0:xrvr-1(config-if)# destination 10.1.1.10
RP/0/0/CPU0:xrvr-1(config-if)# path-option 1 explicit name PATH3_BSID segment-routing
RP/0/0/CPU0:xrvr-1(config-if)# commit

RP/0/0/CPU0:xrvr-1# show mpls traffic-eng tunnels 1 detail
Name: tunnel-te1  Destination: 10.1.1.10  Ifhandle:0x2f80
  Signalled-Name: xrvr-1_t1
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1, (Segment-Routing) type explicit PATH3_BSID (Basis for Setup)
<...>
 Binding SID: 24002
<...>
    Segment-Routing Path Info (IS-IS 1 level-2)
      Segment0[Node]: 10.1.1.3, Label: 16003
      Segment1[ - ]: Label: 24008
```

The path is a chain of SR-TE policies stitched together using the binding-SIDs, providing a seamless end-to-end path.

```
RP/0/0/CPU0:xrvr-1# traceroute 10.1.1.10
Type escape sequence to abort.
Tracing the route to 10.1.1.10
 1  99.1.2.2 [MPLS: Labels 16003/24008 Exp 0] 29 msec  19 msec  19 msec
 2  99.2.3.3 [MPLS: Label 24008 Exp 0] 29 msec  19 msec  19 msec
 3  99.3.4.4 [MPLS: Labels 24003/16005/24012 Exp 0] 29 msec  19 msec  19 msec
 4  99.4.6.6 [MPLS: Labels 16005/24012 Exp 0] 29 msec  29 msec  19 msec
 5  99.5.6.5 [MPLS: Label 24012 Exp 0] 29 msec  29 msec  19 msec
 6  99.5.9.9 [MPLS: Label 16010 Exp 0] 19 msec  19 msec  19 msec
 7  99.9.10.10 29 msec  19 msec  19 msec
```