

Configure Segment Routing Path Computation Element

The Segment Routing Path Computation Element (SR-PCE) provides stateful PCE functionality by extending the existing IOS-XR PCEP functionality with additional capabilities. SR-PCE is supported on the MPLS data plane and IPv4 control plane.

- About SR-PCE, on page 1
- Usage Guidelines and Limitations, on page 2
- Configure SR-PCE, on page 2
- PCE override of SR-TE policies, on page 7
- PCE-Initiated SR Policies, on page 15
- SR-PCE Flexible Algorithm Multi-Domain Path Computation, on page 17
- Exclude Network Resources during Path Computation over SR-TE Policies, on page 22
- Configure the Shortest Path for Disjoint Candidate Paths, on page 27
- Enable Strict Disjointness for SR-TE policies in the PCE, on page 29
- ACL Support for PCEP Connection, on page 31
- Inter-Domain Path Computation Using Redistributed SID, on page 31
- PCE Support for MPLS-TE LSPs, on page 34
- Configuring the North-Bound API on SR-PCE, on page 37

About SR-PCE

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.

SR-PCE learns topology information by way of IGP (OSPF or IS-IS) or through BGP Link-State (BGP-LS).

SR-PCE is capable of computing paths using the following methods:

- TE metric—SR-PCE uses the TE metric in its path calculations to optimize cumulative TE metric.
- IGP metric—SR-PCE uses the IGP metric in its path calculations to optimize reachability.

- LSP Disjointness—SR-PCE uses the path computation algorithms to compute a pair of disjoint LSPs. The disjoint paths can originate from the same head-end or different head-ends. Disjoint level refers to the type of resources that should not be shared by the two computed paths. SR-PCE supports the following disjoint path computations:
  - Link – Specifies that links are not shared on the computed paths.
  - Node – Specifies that nodes are not shared on the computed paths.
  - SRLG – Specifies that links with the same SRLG value are not shared on the computed paths.
  - SRLG-node – Specifies that SRLG and nodes are not shared on the computed paths.

  When the first request is received with a given disjoint-group ID, the first LSP is computed, encoding the shortest path from the first source to the first destination. When the second LSP request is received with the same disjoint-group ID, information received in both requests is used to compute two disjoint paths: one path from the first source to the first destination, and another path from the second source to the second destination. Both paths are computed at the same time.

TCP Authentication Option

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.

Usage Guidelines and Limitations

To ensure PCEP compatibility, we recommend that the Cisco IOS XR version on the SR-PCE be the same or later than the Cisco IOS XR version on the PCC or head-end.

Configure SR-PCE

This task explains how to configure SR-PCE.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters mode.
Step 2	**pce** **Example:** `RP/0/RP0/CPU0:router(config)# pce`	Enables PCE and enters PCE configuration mode.
Step 3	**address ipv4** *address* **Example:** `RP/0/RP0/CPU0:router(config-pce)# address ipv4 192.168.0.1`	Configures a PCE IPv4 address.
Step 4	**state-sync ipv4** *address* **Example:** `RP/0/RP0/CPU0:router(config-pce)# state-sync ipv4 192.168.0.3`	Configures the remote peer for state synchronization.
Step 5	**tcp-buffer size** *size* **Example:** `RP/0/RP0/CPU0:router(config-pce)# tcp-buffer size 1024000`	Configures the transmit and receive TCP buffer size for each PCEP session, in bytes. The default buffer size is 256000. The valid range is from 204800 to 1024000.
Step 6	**password** {**clear** | **encrypted**} *password* **Example:** `RP/0/RP0/CPU0:router(config-pce)# password encrypted pwd1`	Enables 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. **Note** TCP-AO and TCP MD5 are never permitted to be used simultaneously.
Step 7	**tcp-ao** *key-chain* [**include-tcp-options**] [**accept-ao-mismatch-connection**] **Example:** `RP/0/RP0/CPU0:router(config-pce)# tcp-ao`	Enables 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.

	Command or Action	Purpose
	`pce_tcp_ao include-tcp-options`	• **include-tcp-options**—Includes other TCP options in the header for MAC calculation. • **accept-ao-mismatch-connection**—Accepts connection even if there is a mismatch of AO options between peers. **Note** TCP-AO and TCP MD5 are never permitted to be used simultaneously.
Step 8	**segment-routing** {**strict-sid-only** | **te-latency**} **Example:** `RP/0/RP0/CPU0:router(config-pce)# segment-routing strict-sid-only`	Configures the segment routing algorithm to use strict SID or TE latency. **Note** This setting is global and applies to all LSPs that request a path from this controller.
Step 9	**timers** **Example:** `RP/0/RP0/CPU0:router(config-pce)# timers`	Enters timer configuration mode.
Step 10	**keepalive** *time* **Example:** `RP/0/RP0/CPU0:router(config-pce-timers)# keepalive 60`	Configures the timer value for locally generated keep-alive messages. The default time is 30 seconds.
Step 11	**minimum-peer-keepalive** *time* **Example:** `RP/0/RP0/CPU0:router(config-pce-timers)# minimum-peer-keepalive 30`	Configures the minimum acceptable keep-alive timer that the remote peer may propose in the PCEP OPEN message during session establishment. The default time is 20 seconds.
Step 12	**reoptimization** *time* **Example:** `RP/0/RP0/CPU0:router(config-pce-timers)# reoptimization 600`	Configures the re-optimization timer. The default timer is 1800 seconds.
Step 13	**exit** **Example:**	Exits timer configuration mode and returns to PCE configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-pce-timers)# exit`	

Configure the Disjoint Policy (Optional)

This task explains how to configure the SR-PCE to compute disjointness for a pair of LSPs signaled by PCCs that do not include the PCEP association group-ID object in their PCEP request. This can be beneficial for deployments where PCCs do not support this PCEP object or when the network operator prefers to manage the LSP disjoint configuration centrally.

Procedure

	Command or Action	Purpose
Step 1	**disjoint-path** **Example:** `RP/0/RP0/CPU0:router(config-pce)# disjoint-path`	Enters disjoint configuration mode.
Step 2	**group-id** *value* **type** {**link** | **node** | **srlg** | **srlg-node**} [**sub-id** *value*] **Example:** `RP/0/RP0/CPU0:router(config-pce-disjoint)# group-id 1 type node sub-id 1`	Configures the disjoint group ID and defines the preferred level of disjointness (the type of resources that should not be shared by the two paths): • **link**—Specifies that links are not shared on the computed paths. • **node**—Specifies that nodes are not shared on the computed paths. • **srlg**—Specifies that links with the same SRLG value are not shared on the computed paths. • **srlg-node**—Specifies that SRLG and nodes are not shared on the computed paths. If a pair of paths that meet the requested disjointness level cannot be found, then the paths will automatically fallback to a lower level: • If the requested disjointness level is SRLG or node, then link-disjoint paths will be computed.

	Command or Action	Purpose
		• If the requested disjointness level was link, or if the first fallback from SRLG or node disjointness failed, then the lists of segments encoding two shortest paths, without any disjointness constraint, will be computed.
Step 3	**strict** **Example:** `RP/0/RP0/CPU0:router(config-pce-disjoint)# strict`	(Optional) Prevents the automatic fallback behavior of the preferred level of disjointness. If a pair of paths that meet the requested disjointness level cannot be found, the disjoint calculation terminates and no new path is provided. The existing path is not modified.

Step 4	**lsp** {**1** | **2**} **pcc ipv4** *address* **lsp-name** *lsp_name* [**shortest-path**]
	Example:
	`RP/0/RP0/CPU0:router(config-pce-disjoint)# lsp 1 pcc ipv4 192.168.0.1 lsp-name rtrA_t1 shortest-path`
	`RP/0/RP0/CPU0:router(config-pce-disjoint)# lsp 2 pcc ipv4 192.168.0.5 lsp-name rtrE_t2`
		Adds LSPs to the disjoint group.
		The **shortest-path** keyword forces one of the disjoint paths to follow the shortest path from the source to the destination. This option can only be applied to the the first LSP specified.

Global Maximum-delay Constraint

This feature allows a PCE to compare the cumulative latency of a computed path against a global maximum-delay constraint value. If the latency of the computed path exceeds this global constraint, the path is not considered valid. This ensures that all latency-based paths computed by the PCE and signaled to the PCCs in the network do not exceed this maximum-delay constraint.

```
pce
 constraints
   bounds
    cumulative
     type
      latency <1-4294967295> Bound metric value in microseconds
```

Configuration

To configure a PCE for specifying maximum cumulative latency metric, you must complete the following configurations:

```
RP/0/RSP0/CPU0:ios(config)# pce
RP/0/RSP0/CPU0:ios(config-pce)# constraints
RP/0/RSP0/CPU0:ios(config-pce-constr)# bounds
RP/0/RSP0/CPU0:ios(config-pce-constr-bounds)# cumulative
RP/0/RSP0/CPU0:ios(config-pce-constr-bounds-type)# type latency 1000000
RP/0/RSP0/CPU0:ios(config-pce-constr-bounds-type)#
```

Verification

Verify using the **show** command:

```
RP/0/RSP0/CPU0:ios(config-pce-constr-bounds-type)# show
Wed Oct 12 22:18:22.962 UTC
pce
 constraints
  bounds
   cumulative
    type latency 1000000
   !
  !
 !
!
```

PCE override of 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 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 – An unsigned non-zero 32-bit integer value that associates the SR Policy with an intent or objective (e.g., low latency).
- End-point – The destination of the SR-TE policy.

There are fewer PCEs in the network compared to the PCCs. However, PCE versions may be newer than the PCC version because it is easy to update the lesser PCEs than the higher number of PCCs. This ensures PCE supports more features with the latest Internet Assigned Numbers Authority (IANA) code points than PCCs.

IOS-XR PCE implementation is based on the PCEP standards and the implementations of the PCEP vary leading to interoperability challenges. To overcome these challenges, you can configure PCE with the override-rules (OVR). These OVRs have a filter part and an overriding part. The matching criteria filter matches all the (Segment Routing) SR policies against these OVRs, suppose they match, then the overriding part applies to these SR policies attributes.

For example, suppose that a PCC is outdated and does not support reporting the latency metric type. In that case, PCC reports to the LSP with the metric-type that it supports and the PCE applies the OVRs that you configure to update the metric-type upon reception.

Usage guidelines

Some usage guidelines that apply to the PCE override of SR-TE policies feature.

- Lower sequence numbers have higher precedence. LSP can be matched to one override sequence at a time. It will be matched to the highest preference sequence and further matching stops.

- Show commands such as “show pce lsp” display values received from PCC, so applied changes are visible only in values, which are based on path-computation.
- In a PCE High Availability case, where there are multiple PCEs for a particular PCC, configure this feature likewise on all PCEs, to allow seamless transition of delegation between the PCEs.
- PCE (North Bound) NB-API clients only see overridden attributes in their communication with PCEs.
- The PCE OVR feature is seamless for PCCs. This means that PCCs will not know if the tunnel’s attributes are overridden. PCEs uses overridden attributes internally and sends only the original attributes to the PCCs.

Limitations

- Supports only Segment Routing (SR) setup type.
- Supports only ten override rules.
- Only one color or color range per override rule is supported.
- Does not support SRv6 policies.
- You cannot configure the filtration option "all" under peer and LSP with another filtration option under the same config submode. For example, you cannot provide all and regex together under the LSP matching criteria.
- Only IPv4 Access Control Lists (ACL)s are supported for matching PCEP peers.
  - If the ACL is empty or there are no entries within the ACL, then it is implicit permit.
  - If the ACL is not configured but the name is specified, then it is implicit permit.
  - In all other cases, if no entries are matched in the ACL, then it is implicit deny.
  - The behavior aligns with ACL behavior in other areas on an XR router.

Prerequisites

Before you configure the PCE override of SR-TE policies, you can use the **try-regex** utility to test the regexes. This utility is an optional step and does not alter the system.

This example shows how to verify if the

override-rules sequence *sequence-number* [**matching-criteria lsp** { **all** | **name** *lsp-name-regex* }] works:

```
Router# pce try-regex ^cp_c_[0-9]+$ cp_c_5000
Regex verification utility
Regex:       ^cp_c_[0-9]+$
Test string: cp_c_5000
Result:      Matched
```

PCE override of PCC-initiated SR-TE policies

The Path Computation Element (PCE) is a key component of Software-Defined Networking (SDN) for Traffic Engineering (TE), which can dynamically compute and recompute paths for Segment Routing Traffic Engineering (SR-TE) policies. These computations are guided by specified path optimization metrics and constraints. By leveraging the PCE override feature, the PCE can modify certain attributes or constraints of these policies, enabling support for a broader range of use cases and enhanced functionality.

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
PCE override of PCC-initiated policies	Release 7.7.1	You can now override the LSP attributes in the PCEs based on the matching criteria of the path computation element protocol (PCEP) peer and LSP attributes. This feature improves the path computation of the PCC-initiated SR-TE policies. You can configure the PCE override rule for one, many, or all PCCs. This feature introduces the following commands: • **override-rules (PCE)** • **pce try-regex** • **show pce override-rules detail**

Matching of SR-TE policies is done before PCInitiate is sent to PCC to allow modification of data included in messages sent to PCC before LSP is created.

The LSP matching is not re-evaluated again after receiving the next PCEP report message. Based on these matching criteria, you can override the values of the LSP attributes.

Configure PCE override of PCC-initiated SR-TE policies

This configuration defines an **override rule** for a PCE on a network router, which allows the PCE to modify certain attributes of Label Switched Paths (LSPs) that meet specific criteria for PCC-initiated SR-TE policies.

Procedure

Step 1 **configure**

Example:

```
Router# configure
```

Enters the global configuration mode.

Step 2 **pce**

Example:

```
Router(config)# pce
```

Enters the PCE configuration mode.

Step 3 **override-rules**

Example:

```
Router(config-pce)# override-rules
```

Enters the override rules submode.

Step 4 **sequence** *value*

Example:

```
Router(config-pce-ovr-rule)# sequence 100
```

Creates a rule with the sequence number 100.

Step 5 **matching-criteria**

Example:

```
Router(config-pce-ovr-rule-seq)# matching-criteria
```

Specifies a matching criteria. For example, **peer all**, **lsp all**, and so on.

Step 6 **peer all**

Example:

```
Router(config-pce-ovr-rule-crit)# peer
Router(config-pce-ovr-rule-peer)# all
Router(config-pce-ovr-rule-peer)# exit
```

Specifies that this rule applies to all peers.

.

Step 7 **lsp all** *range*

Example:

```
Router(config-pce-ovr-rule-crit)# lsp all
Router(config-pce-ovr-rule-crit)# exit
```

Specifies all LSPs in the PCE tunnel database.

Step 8 **colors** *range*

Example:

```
Router(config-pce-ovr-rule-crit)# colors 0-50
Router(config-pce-ovr-rule-crit)# exit
```

Defines that the colors in the 0 to 50 range to match this rule.

Step 9 **override metric type** *metric-type*

Example:

```
Router(config-pce-ovr-rule-seq)# override metric type igp
```

Specifies that the IGP (Interior Gateway Protocol) metric for matching LSPs is overridden by the PCE.

Step 10 **override constraints bandwidth** *value*

Example:

```
Router(config-pce-ovr-rule-seq)# override constraints bandwidth 1000
```

Specifies that the bandwidth constraint for matching LSPs is overridden and set to 1000.

Step 11 **commit**

Example:

```
Router(config-pce-ovr-rule-seq)# commit
```

Saves and apply these configuration changes.

Running configuration

These are the running configurations for the PCE override for the PCC-initiated SR-TE policies.

- Match all LSPs from all peers and modify metric type to latency:

```
pce
 override-rules
   sequence 1
    matching-criteria
     peer
      all
     lsp
      all

 override
     metric
      type latency
!
```

- Match LSP name using regex and peer based on provided ACL and change metric type to IGP:

```
pce
 override-rules
   sequence 1
    matching-criteria
     peer
      access-list ipv4 PCC1
    !
     lsp
      name ^cfg_test2_.*$
   !
   override
    metric
     type igp
  !
```

- Match policy using specified Segment Routing (SR) policy color from all peers and change bandwidth to 1000 kbps:

```
pce
 override-rules
   sequence 1
    matching-criteria
     peer
      all
    !
     lsp
       colors 10
   !
    override
```

```
        constraints
         bandwidth 1000
    !
```

Verification

Show command displays operational values, which are applied during the override. For example, a bandwidth value is converted into IEEE float format. The value that is displayed in the show command may not be equal to the value configured, because of the limited precision of IEEE format used in PCEP.

Verify using the show command **show pce override-rules [sequence <sequence>]**

This is the show command output with two override rules that are configured and each matched one LSP are:

```
Router# show pce override-rules details

PCE's Override Rule database:
------------------- ----------
Sequence number: 1
 Matching criteria:
  Peer:
   IPv4 ACL name: PCC1
  LSP
   Regex: ^cfg_test1_.*$
 Override:
  Metric type: Latency
  Constraints:
 Matching LSPs:
  Peer: 192.168.0.1, Tunnel name: cfg_test1_xxx_discr_5

Sequence number: 2
 Matching criteria:
  Peer:
   IPv4 ACL name: PCC1
  LSP
   Regex: ^cfg_test2_.*$
 Override:
  Metric type: IGP
  Constraints:
 Matching LSPs:
  Peer: 192.168.0.1, Tunnel name: cfg_test2_xxx_discr_5
```

PCE override of PCE-initiated SR-TE policies

The PCE override feature can be applied to PCE-initiated SR-TE policies and serves several purposes, including controlling PCEP communication with PCCs to ensure interoperability. This feature is particularly useful in scenarios where specific adjustments are needed to address compatibility issues, allowing seamless integration across diverse network devices and vendors.

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Interoperability between PCEP peers for PCE-initiated SR-TE policies	Release 25.2.1	This feature allows interoperability between PCEP peers for PCE-initiated SR-TE policies by blocking encoding PCEP objects and sub-objects thus ensuring seamless integration across diverse network devices and vendors. This feature applies only to the PCE-initiated SR-TE policies. This feature introduces these changes: **CLI:** New keywords, **pcep**, **advertise**, **block-object**, and **sub-type** were introduced in the **override-rules (PCE)** command.

Configure PCE override of PCE-initiated SR-TE policies

This configuration enables the PCE to adjust PCEP communication with specific PCEP peers (PCCs) and LSPs, effectively addressing interoperability challenges and ensuring seamless integration across diverse network devices and vendors.

Procedure

Step 1 **configure**

Example:

```
Router# configure
```

Enters the global configuration mode.

Step 2 **pce**

Example:

```
Router(config)# pce
```

Enters the PCE configuration mode.

Step 3 **override-rules**

Example:

```
Router(config-pce)# override-rules
```

Enters the override rules submode.

Step 4 **sequence** *value*

Example:

```
Router(config-pce-ovr-rule)# sequence 100
```

Creates a rule with the sequence number 100.

Step 5 **matching-criteria**

Example:

```
Router(config-pce-ovr-rule-seq)# matching-criteria
```

Specifies a matching criteria. For example, **peer all**, **lsp all**, and so on.

Step 6 **peer access-list ipv4***acl-name*

Example:

```
Router(config-pce-ovr-rule-crit)# peer
Router(config-pce-ovr-rule-peer)# access-list ipv4 PCC1
Router(config-pce-ovr-rule-peer)# exit
```

Specifies that this rule applies to IPv4 peers.

.

Step 7 **lsp name** *regex*

Example:

```
Router(config-pce-ovr-rule-crit)# lsp name ^cfg_test2_.*$
Router(config-pce-ovr-rule-crit)# exit
```

Defines that LSP name (specified as a regular expression) match this rule.

Step 8 **override pcep advertise** *value* **sub-type** *value*

Example:

```
Router(config-pce-ovr-rule-seq)# override pcep advertise
```

Overrides the attributes that the PCE advertises to its PCC peers via the PCEP.

Step 9 **block-object** *value* **sub-type** *value*

Example:

```
Router(config-pce-ovr-rule-seq)# block-object 40 sub-type 6
```

Blocks encoding of PCEP ASSOCIATION object with object class 40 and ASSOCIATION type 6 ("SR Policy Association") as described in the Path Computation Element Protocol (PCEP) Numbers.

Note
Currently, **block-object** *40* and **sub-type** *6* are the only supported object and sub-object types.

Step 10 **commit**

Example:

```
Router(config-pce-ovr-rule-seq)# commit
```

Saves and apply these configuration changes.

Running configuration

This is the running configuration for the PCE override for the PCE-initiated SR-TE policies.

Match LSP name using regex and peer based on provided ACL and block the PCEP ASSOCIATION object of association type 6 SR Policy Association in PCInitiate message, which is sent from PCE to PCC to create LSP:

```
pce
 override-rules
   sequence 1
    matching-criteria
     peer
      access-list ipv4 PCC1
    !
     lsp
      name ^cfg_test2_.*$
   !
   override
    pcep
     advertise
      block-object 40
       sub-type 6
    !
   !
 !
```

Verification

This show command output demonstrates an override rule configured to block a PCEP sub-object. This resolves interoperability issues with PCC implementations that do not support the PCEP object.

Note Interoperability with fully capable PCCs does not need this configuration.

```
Router# show pce override-rules details

PCE's Override Rule database:
-------------------- ----------
Sequence number: 1
 Matching criteria:
  Peer:
   IPv4 ACL name: PCC1
  LSP
   Regex: ^cfg_test1_.*$
 Override:
   PCEP advertise:
    Objects blocked:
     Type: 40
      Sub-Type: 6
   Constraints:
```

Note Unlike the PCC-initiated LSPs, the PCE-initiated LSPs are not tracked in `Matching LSPs:` in the show output as matching occurs only after the LSP is created.

PCE-Initiated SR Policies

Use cases based on centralized optimization, such as congestion mitigation solutions, rely on the ability of the PCE to signal and instantiate SR-TE policies in the network. We refer to this as PCE-initiated SR-TE policies.

PCE-initiated SR-TE policies can be triggered via Crossworks Network Controller (recommended approach) or via CLI at the PCE.

For more information on configuring SR-TE policies, see the SR-TE Policy Overview.

The PCE deploys the SR-TE policy using PCC-PCE communication protocol (PCEP).

1. PCE sends a PCInitiate message to the PCC.
2. If the PCInitiate message is valid, the PCC sends a PCRpt message; otherwise, it sends PCErr message.
3. If the PCInitiate message is accepted, the PCE updates the SR-TE policy by sending PCUpd message.

You can achieve high-availability by configuring multiple PCEs with SR-TE policies. If the head-end (PCC) loses connectivity with one PCE, another PCE can assume control of the SR-TE policy.

Configuration Example: PCE-Initiated SR Policy with Explicit SID List

To configure a PCE-initiated SR-TE policy, you must complete the following configurations:

1. Enter PCE configuration mode.
2. Create the segment list.



Note When configuring an explicit path using IP addresses of intermediate links, the SR-TE process prefers the protected Adj-SID of the link, if one is available.

3. Create the policy.

```
/* Enter PCE configuration mode and create the SR-TE segment lists */
Router# configure
Router(config)# pce

/* Create the SR-TE segment lists */
Router(config-pce)# segment-routing
Router(config-pce-sr)# traffic-eng
Router(config-pce-sr-te)# segment-list name addr2a
Router(config-pce-sr-te-sl)# index 10 address ipv4 10.1.1.2
Router(config-pce-sr-te-sl)# index 20 address ipv4 10.2.3.2
Router(config-pce-sr-te-sl)# index 30 address ipv4 10.1.1.4
Router(config-pce-sr-te-sl)# exit

/* Create the SR-TE policy */
Router(config-pce-sr-te)# peer ipv4 10.1.1.1
Router(config-pce-sr-te)# policy P1
Router(config-pce-sr-te-policy)# color 2 end-point ipv4 2.2.2.2
Router(config-pce-sr-te-policy)# candidate-paths
Router(config-pce-sr-te-policy-path)# preference 50
Router(config-pce-sr-te-policy-path-preference)# explicit segment-list addr2a
Router(config-pce-sr-te-pp-info)# commit
Router(config-pce-sr-te-pp-info)# end
Router(config)#
```

Running Config

```
pce
 segment-routing
  traffic-eng
   segment-list name addr2a
    index 10 address ipv4 10.1.1.2
    index 20 address ipv4 10.2.3.2
    index 30 address ipv4 10.1.1.4
   !
   peer ipv4 10.1.1.1
    policy P1
     color 2 end-point ipv4 2.2.2.2
     candidate-paths
      preference 50
       explicit segment-list addr2a
      !
     !
```

SR-PCE Flexible Algorithm Multi-Domain Path Computation

Flexible Algorithm provides a traffic engineered path automatically computed by the IGP to any destination reachable by the IGP. With the SR-PCE Flexible Algorithm Multi-Domain Path Computation feature, SR-PCE can use Flexible Algorithms to compute multi-domain paths. See the Enabling Segment Routing Flexible Algorithm chapter for information about Segment Routing Flexible Algorithm.

The SR-PCE Flexible Algorithm Multi-Domain Path Computation feature incorporates the following functionality:

- BGP-LS has been augmented to allow selected nodes to advertise the Flexible Algorithm definition (FAD) to the SR-PCE
- PCEP has been augmented (vendor-specific object) to allow a PCC to indicate SR policy constraint based on the Flexible Algorithm instance number
- SR-PCE algorithms have been augmented to compute paths based on a Flexible Algorithm constraint

The SR-PCE Flexible Algorithm multi-domain path computation requires the following:

- The same Flexible Algorithm instance ID is used across domains.
- The metric for those Flexible Algorithm instances must be the same across domains.
- The affinity constraints for those Flexible Algorithm instances may be different across domains.
- Multiple Flexible Algorithms can exist in a domain.

For example, considering a multi-domain topology (Domain 1 and Domain 2), the following scenarios meet the requirements listed above:

Scenario	Domain 1	Domain 2
Scenario 1	Flexible Algorithm 128, metric delay	Flexible Algorithm 128, metric delay
Scenario 2	Flexible Algorithm 128, metric delay	Flexible Algorithm 128, metric delay, exclude affinity blue

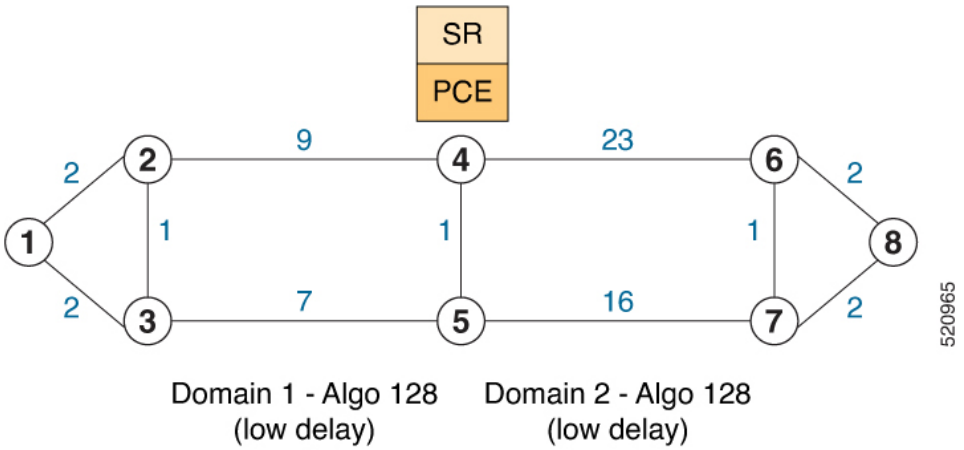
Scenario	Domain 1	Domain 2
Scenario 3	Flexible Algorithm 128, metric delay, exclude affinity yellow	Flexible Algorithm 128, metric delay, exclude affinity blue
Scenario 4	Flexible Algorithm 128, metric delay Flexible Algorithm 129, metric IGP	Flexible Algorithm 128, metric delay Flexible Algorithm 129, metric IGP

Note The use of a Flexible Algorithm constraint in a multi-domain SR topology does not preclude the use of an SR policy that are optimized for a particular metric type. For example, a policy can request a PCE for a Multi Domain policy based on metric delay. SR-PCE computes the path and encodes it with regular prefix SIDs and Adj-SIDs as required. Alternatively, a policy can request to have a constraint for a Flexible Algorithm instance X, which is defined in multiple domains and it minimizes based on metric delay. In this case, the SR-PCE computes the multi-domain path and encodes it using only Flexible Algorithm prefix SIDs. This case benefits from the optimized label stack size that Flexible Algorithm provides (1 label per domain).

Example: SR-PCE Flexible Algorithm Multi-Domain Path Computation Use Case

The following use case depicts a multi-domain topology with two IS-IS processes, each with a Flexible Algorithm instance of 128 that minimizes metric delay. A multi-domain SR policy programmed at Node 1 leverages a Flexible Algorithm 128 path computed by the SR-PCE toward Node 8.

Figure 1: Multi-Domain Topology



Configuration on Node 8

IS-IS and Flexible Algorithm Configuration

```
router isis 2
 is-type level-2-only
 net 49.0002.0000.0000.0008.00
 distribute link-state
 flex-algo 128
```

```
  metric-type delay
  advertise-definition

 address-family ipv4 unicast
  metric-style wide
  router-id 10.1.1.8
  segment-routing mpls
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
   prefix-sid absolute 16008
   prefix-sid algorithm 128 absolute 16808
 !
```

Configuration on Node 4 (ABR/ASBR)

IS-IS and Flexible Algorithm Configuration

```
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0004.00
 distribute link-state instance-id 100
 flex-algo 128
   metric-type delay
   advertise-definition

 address-family ipv4 unicast
   metric-style wide
   router-id 10.1.1.4
   segment-routing mpls
  !
 interface Loopback0
   passive
   address-family ipv4 unicast
    prefix-sid absolute 16004
    prefix-sid algorithm 128 absolute 16804
 !
router isis 2
 is-type level-2-only
 net 49.0002.0000.0000.0004.00
 distribute link-state instance-id 200
 flex-algo 128
   metric-type delay
   advertise-definition

 address-family ipv4 unicast
   metric-style wide
   router-id 10.1.1.4
   segment-routing mpls
  !
 interface Loopback0
   passive
   address-family ipv4 unicast
    prefix-sid absolute 16004
    prefix-sid algorithm 128 absolute 16804
 !
```

BGP-LS Configuration

```
router bgp 65000
 bgp router-id 10.1.1.4
```

```
 address-family link-state link-state
 !
 neighbor-group AS65000-LS-group
  remote-as 65000
  update-source Loopback0
  address-family link-state link-state
  !
 !
 neighbor 10.1.1.10
  use neighbor-group AS65000-LS-group
  description *** To SR-PCE ***
  !
 !
!
```

Configuration on Node 1

IS-IS and Flexible Algorithm Configuration

```
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0001.00
 distribute link-state
 flex-algo 128
   metric-type delay
   advertise-definition

 address-family ipv4 unicast
   metric-style wide
   router-id 10.1.1.1
   segment-routing mpls
  !
 interface Loopback0
   passive
   address-family ipv4 unicast
    prefix-sid absolute 16001
    prefix-sid algorithm 128 absolute 16801
 !
```

SR Policy Configuration

```
segment-routing
 traffic-eng
  policy FOO
   color 100 end-point ipv4 10.1.1.8
   candidate-paths
    preference 100
     dynamic
      pcep
      !
     !
     constraints
      segments
       sid-algorithm 128
      !
     !
    !
   !
  !
 !
!
```

PCC Configuration

```
segment-routing
 traffic-eng
  pcc
   source-address ipv4 10.1.1.1
   pce address ipv4 10.1.1.10
    precedence 10
   !
   report-all
  !
 !
!
```

Configuration on PCE

```
pce
 address ipv4 10.1.1.10
 rest
 !
!
router bgp 65000
 bgp router-id 10.1.1.10
 address-family link-state link-state
 !
 neighbor-group AS65000-LS-group
  remote-as 65000
  update-source Loopback0
  address-family link-state link-state
  !
 !
 neighbor 10.1.1.4
  use neighbor-group AS65000-LS-group
  description *** To Node-4 ***
  !
 !
 neighbor 10.1.1.5
  use neighbor-group AS65000-LS-group
  description *** To Node-5 ***
  !
 !
!
```

Exclude Network Resources during Path Computation over SR-TE Policies

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
Exclude Network Resources during Path Computation over SR-TE Policies	Release 24.1.1	

Feature Name	Release Information	Feature Description
		You can reduce the risk of attacks due to less secure network resources and IP addresses, improve network performance affected by overloaded or congested paths, and ensure higher levels of network stability and availability that could be impacted by resources experiencing issues and requiring maintenance. These improvements are possible because you can now exclude specific network resources using their IP addresses during path computation for traffic over SR-TE policies. Previously, you could not exclude network resources during path computation. The feature introduces these changes: **CLI:** • The **resources** option is introduced in the **segment-routing traffic-eng policy** and **segment-routing traffic-eng on-demand color** commands. • The **resource-list** keyword is introduced in the **segment-routing traffic-eng** command. **YANG Data Models:** • `Cisco-IOS-XR-infra-xtc-oper.yang` • `Cisco-IOS-XR-infra-xtc-agent-oper.yang` • `Cisco-IOS-XR-infra-xtc-agent-cfg.yang` See (GitHub, Yang Data Models Navigator)

You can configure the SR-TE policies as either single or disjoint candidate paths that do not include certain IP addresses or subnets when traffic is routed through the network.

The key benefits of the feature are:

- **Security** - Excluding certain IP addresses or network resources can help protect sensitive resources from unauthorized access or attacks.
- **Performance** - By excluding certain resources, traffic can be directed away from congested parts of the network.
- **Isolation** - Excluding resources can help isolate different parts of the network, preventing issues in one part of the network from affecting others.

Configuration to exclude network resources during Path Computation

To exclude network resources during path computation for traffic, you must complete these high-level tasks in order:

1. Configure the network resources or IP addresses that you want to exclude from the network list.
2. Associate the excluded network resources to candidate paths for SR-TE or ODN SR-TE policies.

Configure the network resources or IP addresses to exclude from the network list

Perform the following task to configure a list of IPv4 addresses that you want to exclude from the network resource list:

```
Router(config)#segment-routing traffic-eng
Router(config-sr-te)#resource-list node_resc_list
Router(config-sr-te-rl)#index 1 ipv4 10.10.10.1
Router(config-sr-te-rl)#index 2 ipv4 10.10.10.8
```

Running Configuration

```
!
segment-routing
 traffic-eng
  resource-list node_resc_list
   index 1 ipv4 10.10.10.1
   index 2 ipv4 10.10.10.8
  !
 !
!
```

Associate the excluded network resources to candidate paths for SR-TE policies

Perform the following task to associate the excluded IPv4 addresses to one or more candidate paths for SR-TE policies:

```
Router(config)#segment-routing traffic-eng
Router(config-sr-te)#policy dynamic_pcep_policy
Router(config-sr-te-policy)#candidate-paths
Router(config-sr-te-policy-path)#preference 100
Router(config-sr-te-policy-path-pref)#constraints resources exclude resource-list
node_resc_list
```

Running Configuration

```
!
segment-routing
 traffic-eng
  policy dynamic_pcep_policy
   candidate-paths
```

```
    preference 100
     constraints
      resources
       exclude resource-list node_resc_list
      !
     !
    !
   !
  !
 !
!
```

Verification

```
Router#show segment-routing traffic-eng policy endpoint ipv4 100.2.1.1 color 8001

Wed Aug 30 22:21:50.014 UTC
SR-TE policy database
---------------------
Color: 8001, End-point: 100.2.1.1
Name: srte_c_8001_ep_100.2.1.1
Status:
Admin: up Operational: up for 00:00:50 (since Aug 30 22:20:59.341)
Candidate-paths:
Preference: 100 (configuration) (active)
Name: dynamic_pcep_policy
Requested BSID: 8001
PCC info:
Symbolic name: cfg_dynamic_pcep_policy_discr_100
PLSP-ID: 14042
Constraints:
Protection Type: protected-preferred
Maximum SID Depth: 12
Exclude Resources: node_resc_list
10.10.10.1
10.10.10.8
Dynamic (pce 100.7.1.1) (valid)
Metric Type: TE, Path Accumulated Metric: 440
SID[0]: 41111 [Adjacency-SID, 101.1.5.1 - 101.1.5.2]
SID[1]: 21600 [Prefix-SID, 100.6.1.1]
SID[2]: 41600 [Adjacency-SID, 101.2.6.2 - 101.2.6.1]
Attributes:
Binding SID: 8001
Forward Class: Not Configured
Steering labeled-services disabled: yes
Steering BGP disabled: no
IPv6 caps enable: yes
Invalidation drop enabled: no
Max Install Standby Candidate Paths: 0
```

Associate the excluded network resources to candidate paths for ODN SR-TE policies

Perform the following task to associate the excluded IPv4 addresses for ODN SR-TE policies:

```
Router(config)#segment-routing
Router(config-sr)#traffic-eng
Router(config-sr-te)#on-demand color 7001
Routerconfig-sr-te-color)#constraints resources exclude resource-list node_resc_list
```

Running Configuration

```
!
segment-routing
 traffic-eng
  on-demand color 7001
```

```
   constraints
    resources
     exclude resource-list node_resc_list
    !
   !
  !
 !
!
```

Verification

```
Router#show segment-routing traffic-eng policy endpoint ipv4 100.2.1.1 color 7001

Wed Aug 30 22:53:01.079 UTC

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

Color: 7001, End-point: 100.2.1.1
  Name: srte_c_7001_ep_100.2.1.1
  Status:
    Admin: up  Operational: up for 00:31:56 (since Aug 30 22:21:04.869)
  Candidate-paths:
    Preference: 200 (BGP ODN) (inactive) (shutdown)
      Requested BSID: dynamic
      Constraints:
        Protection Type: protected-preferred
        Maximum SID Depth: 12
        Exclude Resources: node_resc_list
          10.10.10.1
          10.10.10.8
      Dynamic (inactive)
        Metric Type: IGP,   Path Accumulated Metric: 0
    Preference: 100 (BGP ODN) (active)
      Requested BSID: dynamic
      PCC info:
        Symbolic name: bgp_c_7001_ep_100.2.1.1_discr_100
        PLSP-ID: 14044
      Constraints:
        Protection Type: protected-preferred
        Maximum SID Depth: 12
        Exclude Resources: node_resc_list
          10.10.10.1
          10.10.10.8
      Dynamic (pce 100.7.1.1) (valid)
        Metric Type: IGP,   Path Accumulated Metric: 440
          SID[0]: 41111 [Adjacency-SID, 101.1.5.1 - 101.1.5.2]
          SID[1]: 21600 [Prefix-SID, 100.6.1.1]
          SID[2]: 41600 [Adjacency-SID, 101.2.6.2 - 101.2.6.1]
  Attributes:
    Binding SID: 51679
    Forward Class: Not Configured
    Steering labeled-services disabled: yes
    Steering BGP disabled: no
    IPv6 caps enable: yes
    Invalidation drop enabled: no
    Max Install Standby Candidate Paths: 0
```

Configure the Shortest Path for Disjoint Candidate Paths

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Configure the Shortest Path for Disjoint Candidate Paths	Release 24.1.1	You can now configure the available disjoint paths for Label Switched Paths (LSPs) to prefer the shortest path between two points in the network. This configuration ensures that traffic is routed along the most efficient route in the network. Previously, you could not configure the shortest path preference for disjoint LSPs. The feature introduces these changes: **CLI:** The **shortest-path** keyword is introduced in the **policy candidate-paths constraints disjoint-path** command. **YANG Data Models:** • `Cisco-IOS-XR-infra-xtc-oper.yang` • `Cisco-IOS-XR-infra-xtc-agent-oper.yang` • `Cisco-IOS-XR-infra-xtc-agent-cfg.yang` See (GitHub, Yang Data Models Navigator)

The configuration enables the available disjoint paths to indicate preference for the shortest path between two points in the network. This way the traffic is contained within specific network planes as per the affinity constraints while still provisioning Label Switched Paths (LSPs) that are diverse from each other to meet the path disjointness requirement.

In earlier releases, if path disjointness was configured for LSPs, the Path Computation Element (PCE) prioritized finding disjoint paths for the LSPs over adhering to the affinity constraints.

Configure the shortest path for disjoint candidate paths

Perform the following task to indicate the disjoint path preference for the shortest path in the network:

```
Router(config)#segment-routing traffic-eng
Router(config-sr-te)#policy dynamic_pcep_policy_disjoint
```

```
Router(config-sr-te-policy)#candidate-paths
Router(config-sr-te-policy-path)#preference 100
Router(config-sr-te-policy-path-pref)#constraints disjoint-path group-id 1 type link
shortest-path
```

Running Configuration

```
!
segment-routing
 traffic-eng
   policy dynamic_pcep_policy
   candidate-paths
    preference 100
     constraints
      disjoint-path group-id 1 type link shortest-path

     !
    !
   !
  !
 !
!
```

Verification

```
Router#show pce lsp name cfg_dynamic_pcep_policy_disjoint_discr_100 detail
Output received:
Wed Aug 30 18:33:57.807 UTC

PCE's tunnel database:
----------------------
PCC 100.1.1.1:

Tunnel Name: cfg_dynamic_pcep_policy_disjoint_discr_100
Color: 30115
Interface Name: srte_c_30115_ep_100.2.1.1
 LSPs:
  LSP[0]:
   source 100.1.1.1, destination 100.2.1.1, tunnel ID 9031, LSP ID 24
   State: Admin up, Operation active
   Setup type: Segment Routing
   Binding SID: 30115
   Maximum SID Depth: 12
   Preference: 100
   Bandwidth: requested 0 kbps, applied 0 kbps
   Protection type: protected-preferred
   Prefix-SID algorithm: 0 (set: 0)
   PCEP information:
     PLSP-ID 0x36e9, flags: D:1 S:0 R:0 A:1 O:2 C:0
   LSP Role: Exclude LSP
   State-sync PCE: None
   PCC: 100.1.1.1
   LSP is subdelegated to: None
   Reported path:
     Metric type: TE, Accumulated Metric 30
      SID[0]: Node, Label 21200, Address 100.2.1.1
   Computed path: (Local PCE)
     Computed Time: Wed Aug 30 18:31:12 UTC 2023 (00:02:45 ago)
     Metric type: TE, Accumulated Metric 30
      SID[0]: Node, Label 21200, Address 100.2.1.1
   Reverse path: (Local PCE)
     None
     Computed Time: Wed Aug 30 18:31:12 UTC 2023 (00:02:45 ago)
   Recorded path:
     None
```

```
Disjoint Group Information:
  Type Link-Disjoint, Group 1 (SP)
SR Policy Association Group:
  Color: 30115, Endpoint: 100.2.1.1
  Policy Name: srte_c_30115_ep_100.2.1.1
  Preference: 100
  CP Name: dynamic_pcep_policy_disjoint
```

Enable Strict Disjointness for SR-TE policies in the PCE

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Enable Strict Disjointness for SR-TE Policies in the PCE	Release 24.1.1	You can now enforce strict disjoint constraints for Label Switched Paths (LSPs) and minimize the risk of a single point of failure that affects multiple LSPs. With this feature, if disjoint paths cannot be found for LSPs, the Path Computation Element (PCE) does not return any path. Previously, enforcing strict disjoint constraints for disjoint paths for LSPs was not possible. The feature introduces these changes: **CLI:** The **fallback disable** keyword is introduced in the **segment-routing traffic-eng policy** and **segment-routing traffic-eng on-demand color** commands. **YANG Data Models:** • Cisco-IOS-XR-infra-xtc-oper.yang • Cisco-IOS-XR-infra-xtc-agent-oper.yang • Cisco-IOS-XR-infra-xtc-agent-cfg.yang See (GitHub, Yang Data Models Navigator)

When you enable fallback disable, the Path Computation Element (PCE) does not return any path for LSPs that are not disjoint if there are no disjoint candidate paths. In other words, when the PCE does not find multiple LSPs that satisfy the disjointness requirement, it returns no path. This indicates that there might be a networking issue that prevents the establishment of the desired redundancy, and you can investigate and address the underlying problem.

Without the option, the PCE can relax disjointness by applying an objective function or use a local policy when no objective function is requested.

Enforcing strict disjointness ensures that LSPs do not share any common links or nodes along their paths, minimizing the risk of a single point of failure affecting multiple LSPs. The feature is critical to maintain network reliability and ensures that traffic is properly routed in the event of a network failure.

Configure Strict Disjointness in the PCE

Enable strict disjointness for ODN SR-TE policies

Perform the following steps to enable strict disjointness for ODN SR-TE policies:

```
Router(config)#segment-routing traffic-eng
Router(config-sr-te)#on-demand color 4
Router(config-sr-te-color)#dynamic
Router(config-sr-te-color-dyn)#disjoint-path group-id 1 type node fallback disable
Router(config-sr-te-color-dyn)#commit
```

Running Configuration

```
segment-routing
 traffic-eng
  on-demand color 4
   dynamic
    disjoint-path group-id 1 type node fallback disable
   !
  !
 !
!
```

Enable strict disjointness for SR-TE policies

Perform the following steps to enable strict disjointness for SR-TE policies:

```
Router(config)#segment-routing traffic-eng
Router(config-sr-te)#policy foo
Router(config-sr-te-policy)#color 1 end-point ipv4 10.10.10.1
Router(config-sr-te-policy)#candidate-paths preference 100
Router(config-sr-te-policy-path-pref)#constraints disjoint-path group-id 1 type node fallback
 disable
Router(config-sr-te-policy-path-pref)#commit
```

Running Configuration

```
segment-routing
 traffic-eng
  policy foo
   color 1 end-point ipv4 10.10.10.1
   candidate-paths
    preference 100
     dynamic
      pcep
      !
      metric
       type latency
      !
     !
     constraints
      disjoint-path group-id 1 type node fallback disable
     !
```

```
    !
   !
  !
 !
!
```

Verification

```
Router#sh pce association
Wed Mar 13 14:38:27.173 PDT

PCE's association database:
----------------------
Association: Type Node-Disjoint, Group 1, Strict
 Associated LSPs:
  LSP[0]:
   PCC 10.10.10.1, tunnel name cfg_foo_discr_100,  PLSP ID 4, tunnel ID 8, LSP ID 2,
Configured on PCC
  LSP[1]:
   PCC 10.10.10.1, tunnel name bgp_c_4_ep_10.10.10.2_discr_100,  PLSP ID 1, tunnel ID 5,
LSP ID 3, Configured on PCC
 Status: Satisfied
```

ACL Support for PCEP Connection

PCE protocol (PCEP) (RFC5440) is a client-server model running over TCP/IP, where the server (PCE) opens a port and the clients (PCC) initiate connections. After the peers establish a TCP connection, they create a PCE session on top of it.

The ACL Support for PCEP Connection feature provides a way to protect a PCE server using an Access Control List (ACL) to restrict IPv4 PCC peers at the time the TCP connection is created based on the source address of a client. When a client initiates the TCP connection, the ACL is referenced, and the client source address is compared. The ACL can either permit or deny the address and the TCP connection will proceed or not.

Refer to the Implementing Access Lists and Prefix Lists chapter in the *IP Addresses and Services Configuration Guide* for detailed ACL configuration information.

To apply an ACL to the PCE, use the **pce peer-filter ipv4 access-list** *acl_name* command.

The following example shows how to configure an ACL and apply it to the PCE:

```
pce
 address ipv4 10.1.1.5
 peer-filter ipv4 access-list sample-peer-filter
!
ipv4 access-list sample-peer-filter
 10 permit ipv4 host 10.1.1.6 any
 20 permit ipv4 host 10.1.1.7 any
 30 deny ipv4 any any
!
```

Inter-Domain Path Computation Using Redistributed SID

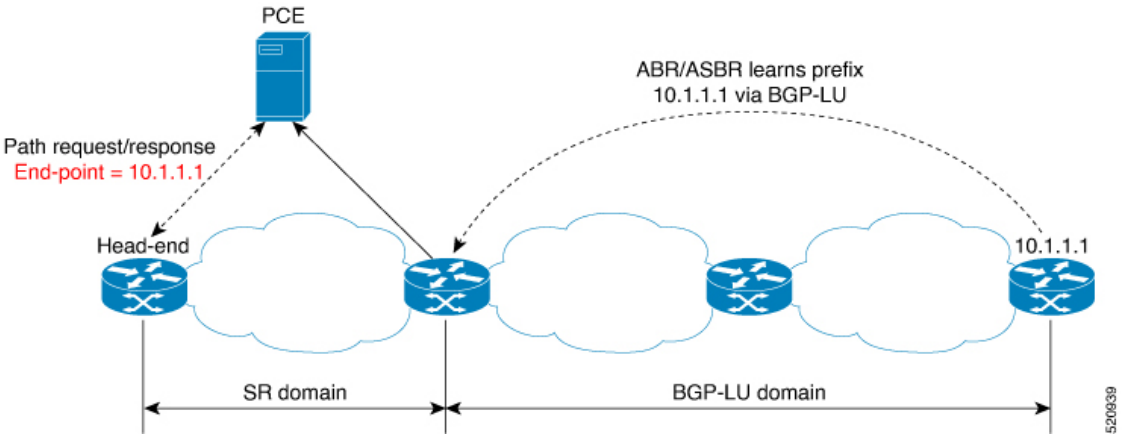
A Path Computation Element (PCE) computes SR-TE paths based on SR topology database that stores connectivity, state, and TE attributes of SR network nodes and links. BGP Labeled Unicast (BGP-LU) provides

MPLS transport across IGP boundaries by advertising loopbacks and label binding of impact edge and border routers across IGP boundaries.

This feature adds new functionality to the SR-PCE that enables it to compute a path for remote non-SR end-point device distributed by BGP-LU.

The remote end-point device in the BGP-LU domain is unknown to the SR-PCE. For the SR-PCE to know about the end-point device, the gateway ABR/ASBR learns the end-point prefix via BGP-LU. The prefix is then redistributed to SR-PCE topology database from the gateway ABR/ASBR. SR-PCE then can compute the best path from the head-end device to the selected gateway router.

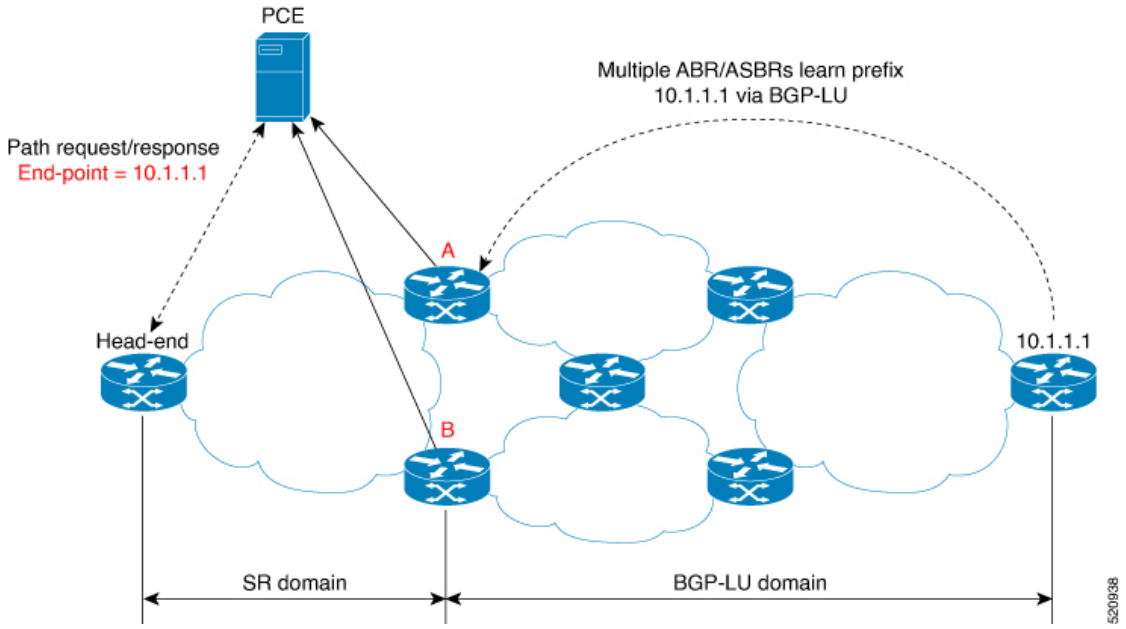
The following topology shows an SR domain and a BGP-LU domain, with a gateway ABR/ASBR between the two domains.



1. The gateway ABR/ASBR is configured with BGP/IGP helper to learn the remote prefix through BGP-LU and redistribute the remote prefix to the IGP helper, then to SR-PCE.
2. The SR-PCE selects the best gateway node to BGP-LU domain and computes the path to reach the remote prefix through the gateway node.
3. The head-end device in the SR domain requests a path to the remote destination and signals the SR profile interworking with the BGP-LU domain.

The BGP-LU prefix advertisement to SR-PCE Traffic Engineer Database (TED) is done by creating an IGP helper on the ABR/ASBR to redistribute BGP-LU prefix information to IGP. IGP then sends the prefix information to the SR-PCE via BGP-LS.

If there are multiple ABR/ASBRs advertising the same remote BGP-LU prefix, the SR-PCE selects the best gateway node to the BGP-LU domain using the accumulative metric from the head-end device to the gateway and the advertised metric from the gateway to the destination.



Example: Inter-Domain Path Computation Using Redistributed SID

The following examples show the configurations for the IGP helper, BGP-LU, and proxy BGP-SR:

Configuration on the End-Point Device

Configure the end-point device to allocate a label for the BGP-LU prefix on the end-point device:

```
router bgp 3107
bgp router-id 1.0.0.8
address-family ipv4 unicast
  network 1.0.0.8/32 route-policy bgplu-com
  allocate-label all

route-policy bgplu-com
  set community (65002:999)
end-policy
```

Configuration on the Gateway ABR/ASBR

1. Configure the remote prefix set and create the route policy for the BGP-LU domain:

```
prefix-set bgplu
  1.0.0.7/32,
  1.0.0.8/32,
  1.0.0.101/32,
  1.0.0.102/32
end-set
!

route-policy bgp2isis
  if destination in bgplu then
    pass
  else
    drop
  endif
```

```
end-policy
!
end
```

2. Configure the helper IGP instance on the Loopback interface:

```
router isis 101
 is-type level-2-only
 net 49.0001.0000.1010.1010.00
 distribute link-state instance-id 9999
 nsf cisco
 nsf lifetime 120
 address-family ipv4 unicast
  metric-style wide
  maximum-paths 64
  router-id Loopback10
  redistribute bgp 3107 metric 200 route-policy bgp2isis
  segment-routing mpls sr-prefer
!
interface Loopback10 >>> this loopback is for gateway SR-TE node-id
  passive
  address-family ipv4 unicast
   prefix-sid index 2001 explicit-null
```

3. Configure the gateway proxy BGP-SR and SR Mapping Server to allocate SR labels:

```
router bgp 3107
 address-family ipv4 unicast
  segment-routing prefix-sid-map
  allocate-label all

segment-routing
global-block 16000 23999
mapping-server
  prefix-sid-map
   address-family ipv4
    1.0.0.7/32 2007
    1.0.0.8/32 2008
    1.0.0.101/32 2101
    1.0.0.102/32 2102
```

PCE Support for MPLS-TE LSPs

This feature allows Cisco's SR-PCE to act as a Path Computation Element (PCE) for MPLS Traffic Engineering Label Switched Paths (MPLS-TE LSPs).

Note For more information about MPLS-TE, refer to the "Implementing MPLS Traffic Engineering" chapter in the *MPLS Configuration Guide* .

The supported functionality is summarized below:

- PCE type: Active Stateful PCE
- MPLS-TE LSP initiation methods:
  - PCE Initiated—An active stateful PCE initiates an LSP and maintains the responsibility of updating the LSP.

- PCC Initiated—A PCC initiates the LSP and may delegate the control later to the Active stateful PCE.

- MPLS-TE LSP metric—Metric optimized by the path computation algorithm:
  - IGP metric
  - TE metric
  - Latency metric
- MPLS-TE LSP constraints—TE LSP attributes to be taken into account by the PCE during path computation:
  - Resource Affinities
  - Path Disjointness
- MPLS-TE LSP parameters:
  - Setup priority—The priority of the TE LSP with respect to taking resources
  - Hold priority—The priority of the TE LSP with respect to holding resources
  - FRR L flag—The "Local Protection Desired" bit. Can be set from an application instantiating an MPLS-TE LSP via SR-PCE. SR-PCE passes this flag to the PCC, and the PCC will enable FRR for that LSP.
  - Signaled Bandwidth—This value can be set from an application instantiating an MPLS-TE LSP via SR-PCE. SR-PCE passes this value to the PCC.
  - Binding SID—A segment identifier (SID) that a headend binds to an MPLS-TE LSP. When the headend receives a packet with active segment (top MPLS label) matching the BSID of a local MPLS-TE LSP, the headend steers the packet into the associated MPLS-TE LSP.

Cisco Crosswork Optimization Engine is an application that leverages the SR-PCE in order to visualize and instantiate MPLS-TE LSPs. For more information, refer to the Visualize SR Policies and RSVP-TE Tunnels chapter in the Cisco Crosswork Optimization Engine 1.2.1 User Guide.

Note No extra configuration is required to enable MPLS-TE support at SR-PCE.

Example: Configuring a PCEP Session (Stateful Mode) on MPLS-TE PCC

The following example shows the configuration for an MPLS-TE PCC to establish a PCEP session with a PCE (IPv4 address 10.1.1.100).

Note MPLS-TE PCC must operate in the stateful PCEP mode when connecting to SR-PCE.

The **instantiation** keyword enables the PCC to support MPLS-TE LSP instantiation by PCE (PCE-initiated).

The **report** keyword enables the PCC to report all the MPLS-TE LSPs configured on that node.

Note PCE-initiated LSPs are automatically reported to all configured PCEs.

The **autoroute-announce** keyword enables autoroute-announce globally for all PCE-initiated LSPs on the PCC.

The **redundancy pcc-centric** keywords enable PCC-centric high-availability model for PCE-initiated LSPs. 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.

```
mpls traffic-eng
 pce
  peer ipv4 10.1.1.100
  !
  stateful-client
   instantiation
   report
   autoroute-announce
   redundancy pcc-centric
  !
 !
!
end
```

Example: Configuring Multiple PCEP Sessions from a PCC Acting as MPLS-TE and SR-TE Headend Toward a Common PCE

The following example shows the configuration for a PCC (IPv4 addresses 10.1.1.1 and 10.1.1.2) to establish two PCEP sessions with a common PCE (IPv4 address 10.1.1.100). One session is configured under MPLS-TE, and the other under SR-TE.

Note The two PCEP sessions must use a different source address on the PCC when connecting to the same PCE.

For more information regarding PCEP configuration at SR-TE PCC, see the *Configure the Head-End Router as PCEP PCC* topic.

```
mpls traffic-eng
 pce
  peer source ipv4 10.1.1.1
  peer ipv4 10.1.1.100
  !
 !
!
end

segment-routing
```

```
 traffic-eng
  pcc
   source-address ipv4 10.1.1.2
   pce address ipv4 10.1.1.100
   !
  !
 !
!
end
```

Configuring the North-Bound API on SR-PCE

Table 6: Feature History Table

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
SR-PCE: Stateful North-Bound API for Tree-SID	Release 7.5.1	The SR-PCE provides a north-bound HTTP-based API to allow communication between the SR-PCE and the Cisco Crosswork Optimization Engine. This release adds stateful north-bound APIs to support real-time monitoring of Tree-SID states on the SR-PCE using a subscription model. For more information, refer to the Cisco Crosswork Optimization Engine User Guides.

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
SR-PCE: North-Bound API for SRv6 and Flexible Algorithm in Cisco Optimization Engine (COE) v3.0 release	Release 7.3.2	The SR-PCE provides a north-bound HTTP-based API to allow communication between the SR-PCE and the Cisco Crosswork Optimization Engine. This release adds support for the following: • Reporting of Flexible Algorithm participation and definitions • SRv6 topology information (nodes, links, Node uSIDs and Adj uSIDs) • SRv6 uSID list and uB6 SIDs allocated for a policy For more information, refer to the Cisco Crosswork Optimization Engine User Guides.

The SR-PCE provides a north-bound HTTP-based API to allow communication between SR-PCE and external clients and applications.

Over this API, an external application can leverage the SR-PCE for topology discovery, SR policy discovery, and SR policy instantiation.

The Cisco Crosswork Optimization Engine is an application that leverages the SR-PCE. For more information, refer to the Cisco Crosswork Optimization Engine User Guides.

Use the following commands under PCE configuration mode to configure the API to allow communication between SR-PCE and external clients or applications.

Command	Description
api authentication {**basic** | **digest**}	Specify the type of authentication: • **basic** – Use HTTP Basic authentication (plaintext) • **digest** – Use HTTP Digest authentication (MD5)
api username *password* {**clear** | **encrypted**} *password*	Add credentials when connecting to API.

Command	Description
api sibling ipv4 *address*	Opens a synchronization channel to another PCE in the same high availability (HA) pair. **Note** For more information regarding SR-PCE HA pairs, refer to the Multiple Cisco SR-PCE HA Pairs chapter of the Cisco Crosswork Optimization Engine 1.2.1 User Guide.

Example: Configuring API on SR-PCE

```
pce
 address ipv4 10.1.1.100
 api
  user admin
   password encrypted 1304131F0202
  !
  authentication digest
  sibling ipv4 10.1.1.200
 !
!
end
```

The following example shows the current active connections:

```
RP/0/0/CPU0:pce1# show tcp brief | i 8080
Thu Aug  6 00:40:15.408 PDT
0xe9806fb8 0x60000000      0      0  :::8080                 :::0                    LISTEN
0xe94023b8 0x60000000      0      0  10.1.1.100:50487        10.1.1.200:8080         ESTAB
0xeb20bb40 0x60000000      0      0  10.1.1.100:8080         10.1.1.200:44401        ESTAB
0xe98031a0 0x60000000      0      0  0.0.0.0:8080            0.0.0.0:0               LISTEN
```

The first and fourth entries show the API server listening for IPv4 and IPv6 connections.

The second and third entries show the established sibling connection between PCE1 (10.1.1.100) and PCE2 (10.1.1.200).