



PLE Service Configuration

- [Private Line Emulation, on page 1](#)
- [How PLE works, on page 2](#)
- [Properties of circuit-style segment routing, on page 3](#)
- [Configure PLE over EVPN-VPWS, on page 6](#)
- [Verify PLE configurations, on page 26](#)

Private Line Emulation

Private Line Emulation (PLE) is a pillar of the Routed Optical Networking solution. PLE allows non-Ethernet type services such as SONET/SDH, OTN, and Fiber Channel to be carried over the same MPLS or Segment Routing network. PLE also supports bit-transparent Ethernet services where required.

High-revenue and legacy private line services, carried over a dedicated inefficient TDM OTN layer, exist in the network infrastructure of most service providers. PLE enables service providers to carry SONET/SDH, OTN, Ethernet, and Fiber Channel over a circuit-style segment routed packet network while maintaining existing service SLAs. PLE utilizes Circuit Emulation (CEM) to transfer PLE client frames over MPLS or SR networks transparently, without changing the characteristics of the original signal.

Benefits of PLE

PLE enables service providers and enterprises to

- collapse network layers
- decrease network complexity, and
- increase network efficiency

Supported payloads

PLE supports these payloads.

Table 1: Supported payloads

| PLE transport type | Supported payloads |
|--------------------|--------------------|
| Ethernet | 1GE and 10GE |

| PLE transport type | Supported payloads |
|--------------------|------------------------------------|
| OTN | OTU2 and OTU2e |
| SONET | OC-48 and OC-192 |
| SDH | STM-16 and STM-64 |
| Fiber Channel | FC1, FC2, FC4, FC8, FC16, and FC32 |

The combination of payloads is also supported.

Supported hardware

PLE supports these hardware:

- NC55-OIP-02 MPA (Modular Port Adapter)

This MPA is supported in the NC55A2-MOD-S and NC57C3-MOD-SYS routers. For more information, see [Specifications of Cisco NCS 5500 series routers](#).

- [Cisco 8010 series router: PLE service endpoint router](#)

How PLE works

Summary

This process ensures the transparent transmission of client Layer 1 frames over a Packet Switched Network, maintaining the integrity and format of the original traffic.

Workflow

These stages describe how the PLE works.

1. Ethernet, OTN, Fiber Channel, SONET, or SDH PLE client traffic is carried on an EVPN-VPWS single-homed service that is created between PLE endpoints.
2. EVPN-VPWS signalling information is carried using BGP between the PLE circuit endpoints either through direct BGP sessions or through a BGP services route-reflector.
3. EVPN-VPWS pseudowire channel is set up between the endpoints when the CEM (Circuit Emulation) client interfaces are configured on each endpoint router and end-to-end transport connectivity using MPLS or Segment Routing transport is enabled.
4. CEM over a packet-switched network (PSN) places the client bit streams into packet payloads with appropriate pseudowire emulation headers.

CEM is a method through which client data can be transmitted over MPLS or Segment Routing networks in a bit-transparent manner, retaining the client Layer 1 frame between sender and receiver.
5. The PLE initiator encapsulates the PLE client traffic and carries it using the EVPN-VPWS service, which runs on MPLS or Segment Routing transport.

- The PLE terminator node extracts the bit streams from the EVPN-VPWS packets and places them onto the PLE client interfaces as defined by the client attribute and CEM profile.

Properties of circuit-style segment routing

Segment routing provides an architecture that caters to both IP-centric transport and connection-oriented transport. IP-centric transport uses the benefits of ECMP and automated protection from Topology-Independent Loop-Free Alternate (TI-LFA).

Connection-oriented transport, which was historically delivered over circuit-switched SONET/SDH networks, must require

- end-to-end bidirectional transport that provides congruent forward and reverse paths, predictable latency, and disjointness
- bandwidth commitment to ensure that there is no impact on the SLA due to network load from other services
- monitoring and maintenance of path integrity with end-to-end 50-millisecond path protection, and
- persistent end-to-end paths regardless of the state of the control plane.

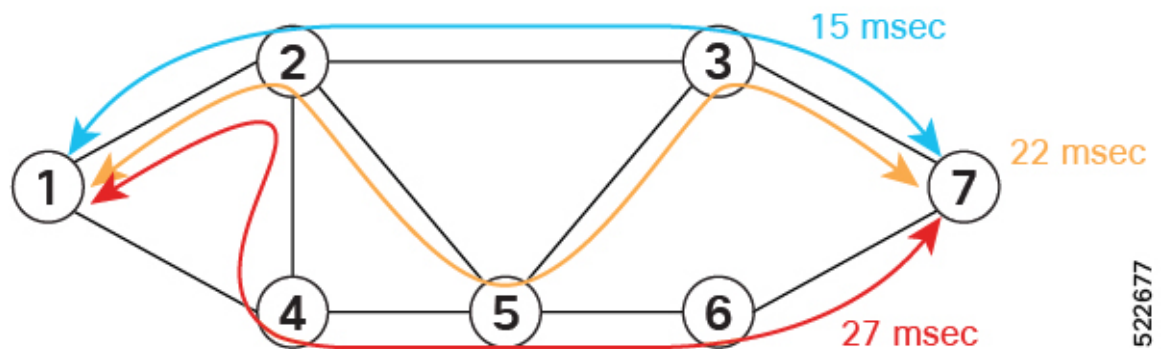
Properties of circuit-style segment routing policies

Circuit-style segment routing (CS-SR) policies provide these properties:

- Guaranteed latency over non-Equal-Cost Multi-Path (ECMP) paths
- Control-plane independent persistency
- Corouted bidirectional path
- Liveness monitoring with path protection switching
- Guaranteed bandwidth

Guaranteed latency over non-ECMP paths

Consider this network with three possible paths from node 1 to node 7. Of the three paths, the best end-to-end delay is provided by the blue path (1 -> 2 -> 3 -> 7). The chosen path is then encoded with adjacency SIDs corresponding to the traversed interfaces to avoid any ECMP. This approach guarantees latency over the path.

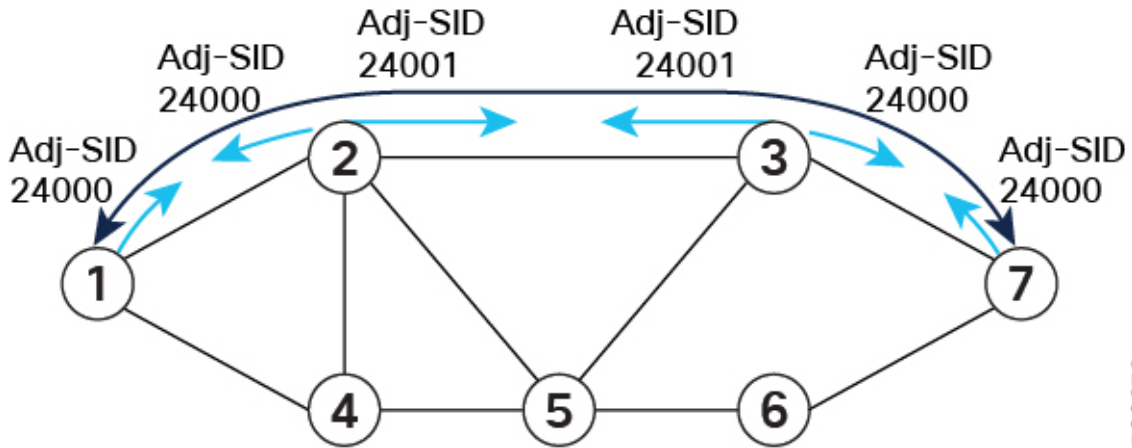


522677

Control-plane independent persistency

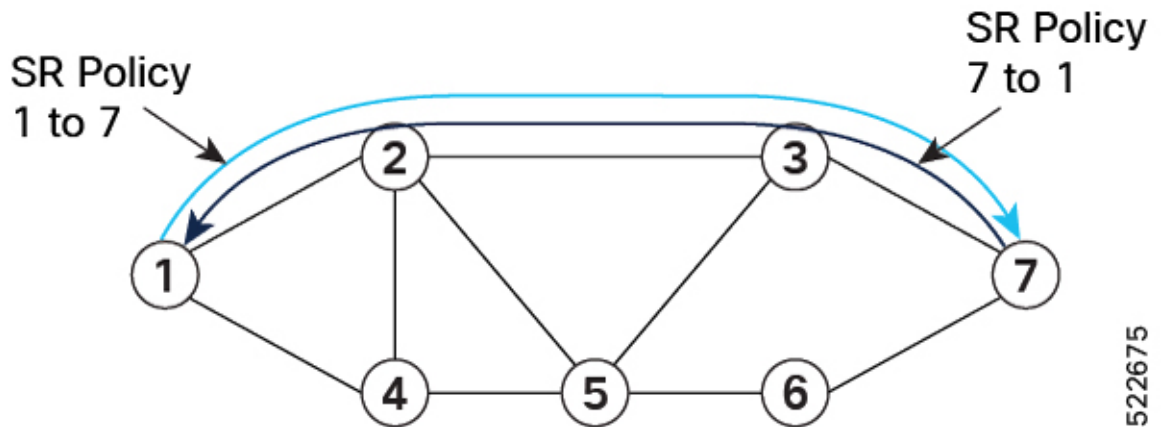
Adjacency SIDs can provide a persistent path independent from control-plane changes such as IGP neighbor flaps and network events such as interface additions, interface flaps, or even the presence of IP on an interface. To achieve this, manually allocate adjacency SIDs to ensure persistent values. For example, this applies after a node reload event. You can also program adjacency SIDs as non-protected to avoid local TI-LFA protection.

The path from node 1 to node 7 is encoded with the segment list of {24000, 24001, 24000} with the adjacency SIDs depicted in this figure. You can encode the path from node 7 to node 1 with the same segment list: {24000, 24001, 24000} by manually allocating the same adjacency SID values for the other direction.



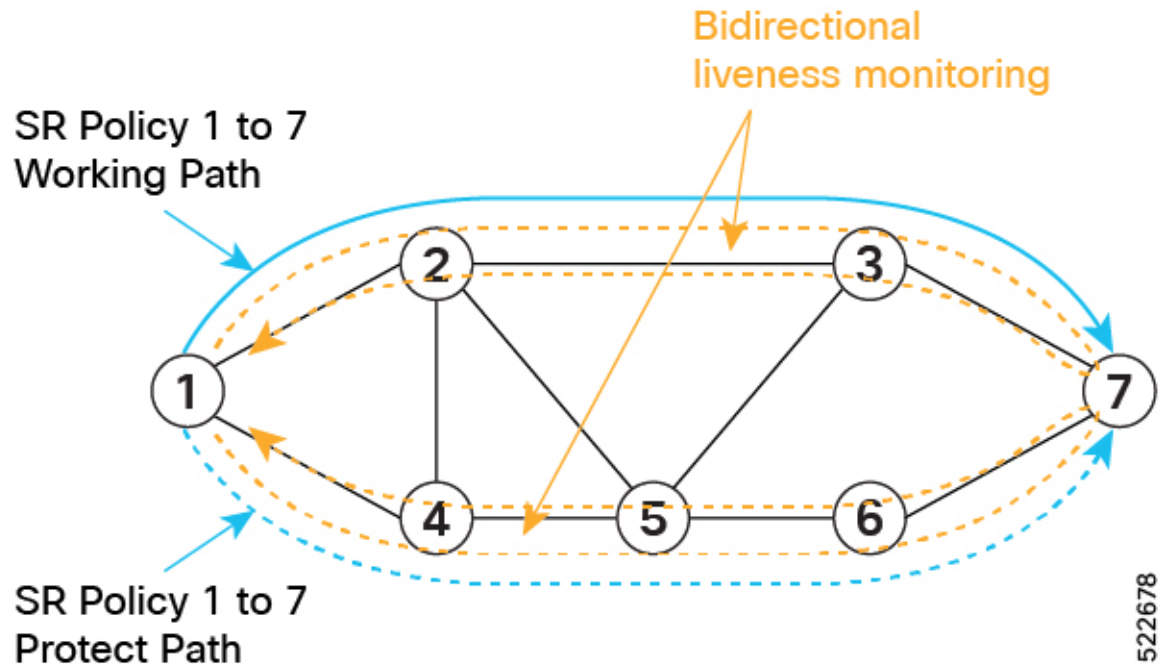
Corouted bidirectional path

Forward and return segment routing policies with congruent paths are routed along the same nodes.



Liveness monitoring with path protection switching

Bidirectional liveness monitoring on the working and protect paths ensures fast and consistent switchover. The protect path is pre-programmed over disjoint facilities.



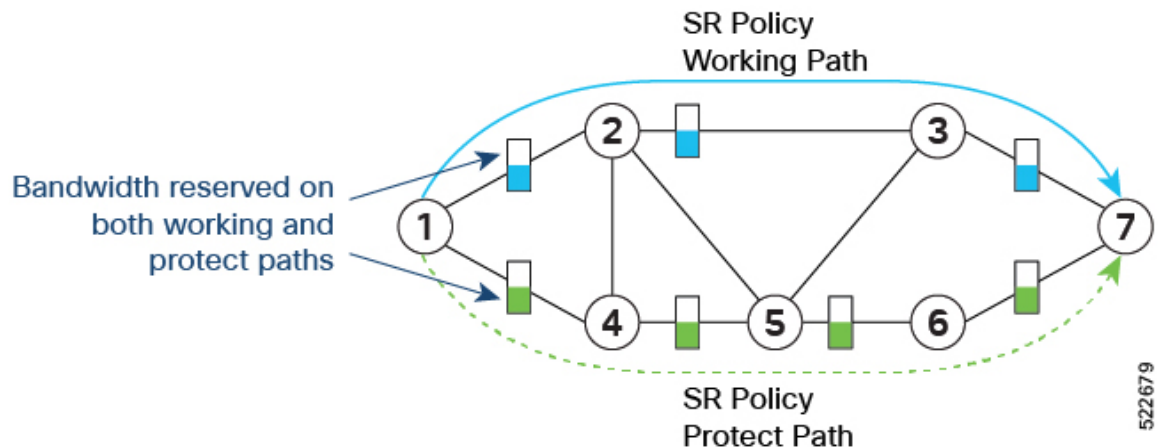
Guaranteed bandwidth

Most services carried over CS-SR policies are constant-rate traffic streams. Any packet loss due to temporary congestion leads to bit errors at the service layer. To prevent this, bandwidth must be tightly managed and guaranteed to the services mapped to CS-SR policies.

A centralized controller manages the bandwidth reservation. The controller maintains the reserved bandwidth on each link based on the requested bandwidth by performing these actions:

- Monitors the amount of traffic
- Uses knowledge of the active path used by the policy
- Computes the per-link reservable bandwidth

Bandwidth is reserved on both the working and protect paths.



Usage guidelines and limitations

- Candidate path (CP) behavior:
 - The working path has the highest CP preference value.
 - The protect path has the second highest CP preference value.
 - The restore path has the lowest CP preference value and is configured as "backup ineligible."
 - Paths with the same role in both directions—working, protect, and restore—must have the same CP preference value.
- Bidirectional path behavior:
 - All paths must be configured as corouted.
 - All paths with the same role in both directions (working, protect, and restore) must have the same bidirectional association ID value.
 - The bidirectional association ID value must be globally unique.

Configure PLE over EVPN-VPWS

Configuring PLE over the EVPN-VPWS service procedure involves these tasks:

1. [Enable frequency synchronization, on page 6](#)
2. [Configure optics controller with payloads, on page 7](#)
3. [Configure a QoS policy, on page 8](#)
4. [Configure CEM interface for L2VPN service, on page 9](#)
5. [Configure performance measurement, on page 9](#)
6. [Configure segment routing adjacency SIDs, on page 10](#)
7. [Define paths between PLE endpoints, on page 11](#)
8. [Configure static circuit-style SR-TE policies, on page 14](#)
9. [Configure PLE over EVPN-VPWS, on page 25](#)

See the [Private Line Emulation over EVPN-VPWS Single Homed](#) section for pre-requisites and other details.

Enable frequency synchronization

Use this task to synchronize clocks between Provider Edge 1 (PE1) and Provider Edge 2 (PE2) routers.

Procedure

- Step 1** Enter these commands to enable frequency synchronization on PLE-PE1.

Example:

```
Router(config)# frequency synchronization
Router(config-freqsync)# quality itu-t option 1
Router(config-freqsync)# exit
Router(config)# interface TwentyFiveGigE0/0/0/24
Router(config-if)# frequency synchronization
Router(config-if-freqsync)# quality transmit exact itu-t option 1 PRC
```

Step 2 Enter the **show frequency synchronization interfaces** command to verify that the clock signal is being transmitted.

Step 3 Enter these commands to enable frequency synchronization on PLE-PE2.

Example:

```
Router(config)# frequency synchronization
Router(config-freqsync)# quality itu-t option 1
Router(config-freqsync)# exit
Router(config)# interface TwentyFiveGigE0/0/0/32
Router(config-if)# frequency synchronization
Router(config-if-freqsync)# selection input
Router(config-if-freqsync)# priority 1
Router(config-if-freqsync)# wait-to-restore 0
```

Step 4 Enter the **show frequency synchronization selection** command to verify that PLE-PE2 is locked to the clock of PLE-PE1.

You have synchronized the clocks between PE1 and PE2 routers.

Configure optics controller with payloads

Use this task to configure the optics controller to support various payload types.

Each port on the NC55-OIP-2 (PID of PLE Modular Port Adapter) can be configured independently for a specific PLE transport type. Use the **port-mode** command under the optics controller to perform this configuration.

For more information about all the supported payloads, see the [Private Line Emulation, on page 1](#) section.

Procedure

Step 1 Enter these commands to configure the optics controller with Ethernet payload.

Example:

```
Router(config)# controller Optics0/0/3/1
Router(config-Optics)# port-mode Ethernet framing cem-packetize rate 10GE
```

The **port-mode** command has several parameters:

- PLE transport type: Choose Ethernet, OTN, SONET, SDH, or Fibre Channel.
- Framing: Set to **cem-packetize** to specify a PLE CEM configuration. This parameter is required.
- Rate: Configure the specific payload.

Step 2 Enter these commands to configure the optics controller with OTN payload.

Example:

```
Router(config)# controller Optics0/0/3/2
Router(config-Optics)# port-mode otn framing cem-packetize rate otu2
```

Step 3 Enter these commands to configure the optics controller with SONET payload.

Example:

```
Router(config)# controller Optics0/0/3/3
Router(config-Optics)# port-mode sonet framing cem-packetize rate OC48
```

Step 4 Enter these commands to configure the optics controller with SDH payload.

Example:

```
Router(config)# controller Optics0/0/3/4
Router(config-Optics)# port-mode sdh framing cem-packetize rate STM16
```

Step 5 Enter these commands to configure the optics controller with Fiber Channel payload.

Example:

```
Router(config)# controller Optics0/0/3/5
Router(config-Optics)# port-mode FC framing cem-packetize rate FC1
```

The optics controller is configured with different payloads.

Configure a QoS policy

Use this task to configure a QoS policy that prioritizes PLE traffic. QoS policy configuration is recommended to prevent high-priority PLE traffic from being dropped during network congestion.

Procedure

Step 1 Enter the **policy-map ple-policy** command to define the parent QoS policy to be applied to the CEM interface to prioritize the PLE traffic.

Example:

```
Router(config)# policy-map ple-policy
```

Step 2 Enter the **class class-default** command to specify the default class.

Example:

```
Router(config)# class class-default
```

A PLE service has only one traffic flow type. Therefore, the default class is used to match all the ingress packets on the PLE CEM interface.

Step 3 Enter the **set traffic-class traffic-class-number** command to specify the traffic class.

Example:

```
Router(config)# set traffic-class 6
```

Cisco NCS 5500 and NCS 5700 platforms use the traffic class to identify specific traffic flows within the router and to determine egress treatment. In this example, ingress traffic is set to traffic class 6. The egress QoS policy matches traffic class 6 to set the appropriate queuing behavior. This value is used as an example; it must be set to the high priority EXP value used across the network. The egress QoS policy is outside the scope of this guide. For details, see *Modular QoS Configuration Guide for Cisco NCS 5500 Series Routers*.

Step 4 Enter the **set mpls experimental topmost** *mpls-exp-value* command to set the topmost MPLS EXP (TC) value.

Example:

```
Router(config)# set mpls experimental topmost 6
```

The MPLS EXP (TC) value can be used at subsequent nodes in the path to classify PLE traffic and set appropriate queuing behavior. This value is used as an example; it must be set to the high priority EXP value used across the network.

You have configured the QoS policy.

Configure CEM interface for L2VPN service

Use this task to configure the CEM interface for the L2VPN EVPN-VPWS service.

Procedure

Step 1 Enter the **interface CEM** command to configure the appropriate CEM interface.

Example:

```
Router(config)# interface CEM0/0/3/1
```

Step 2 Enter the **l2transport** command to enable the CEM interface to be used in a L2VPN EVPN-VPWS service.

Example:

```
Router(config-if)# l2transport
```

Configure performance measurement

Use this task to configure performance measurement for enabling liveness monitoring of the Segment Routing policy.

Procedure

Step 1 Enter the **performance-measurement** command to enter the performance measurement configuration mode.

Example:

```
Router(config)# performance-measurement
```

Step 2 Enter the **liveness-profile name liveness-check** command to create a SR Policy liveness profile.

Example:

```
Router(config-perf-meas)# liveness-profile name liveness-check
```

Step 3 Enter the **liveness-detection multiplier multiplier-value** command to configure liveness detection parameters. This command sets the number of missed liveness probes to determine whether the SR Policy is down.

Example:

```
Router(config-perf-meas)# liveness-detection multiplier 3
```

Step 4 Enter the **probe** command to enter the probe parameter configuration mode.

Example:

```
Router(config-pm-ld-srpolicy)# probe
```

Note

The probe measurement mode is set to loopback by default. In this mode, the probe packets from the sender node are looped back to the sender node to test end-to-end liveness.

Step 5 Enter the **tx-interval interval-value** command to set the interval of probe packets that are sent by the sender.

Example:

```
Router(config-pm-ld-srpolicy-probe)# tx-interval 30000
```

The interval value is set in microseconds. In this example, it is set to 30 seconds.

Note

Using a **tx-interval** value lower than 30 seconds requires hw-offload support. NCS-55A2-MOD supports this in R7.9.1.

Step 6 Enter the **tos dscp dscp-value** command to set the Differentiated Services Code Point (DSCP) value on the probe packets to the specified DSCP.

Example:

```
Router(config-pm-ld-srpolicy-probe)# tos dscp 48
```

In this example, the DSCP value is 48. You must set the DSCP value to the appropriate QoS setting on the provider network so that the liveness probe packets are treated as high priority.

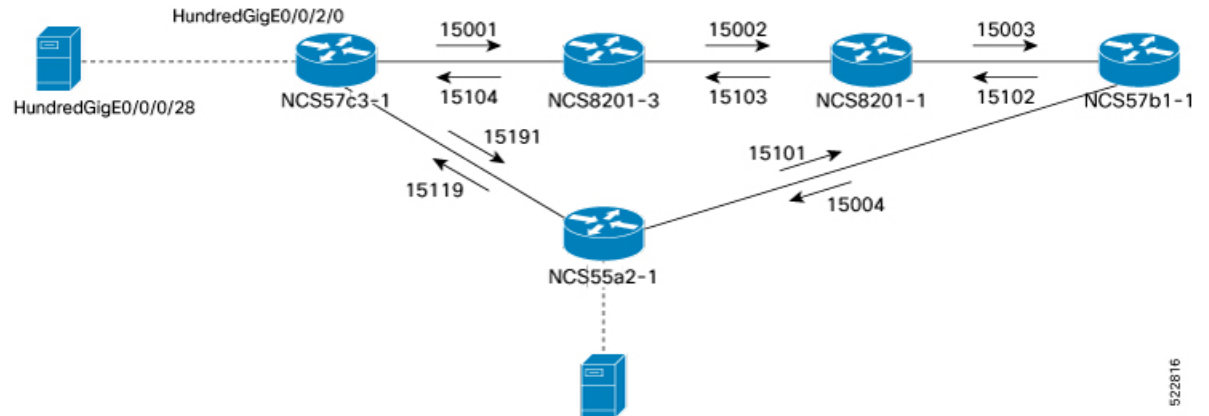
Performance measurement and liveness monitoring are enabled for the Segment Routing policy.

Configure segment routing adjacency SIDs

Adjacency SIDs are unidirectional. They define the explicit path between endpoints. Configure adjacency SIDs on each interface that participates in the SR policy path.

This figure shows the SR adjacency SIDs configured between routers.

Figure 1: Segment routing adjacency SIDs



Use this task to configure segment routing adjacency Segment Identifiers (SIDs), enabling precise control over packet forwarding paths at the link level within a segment routing-enabled network.

Procedure

- Step 1** Enter the **router isis** command to enter the IS-IS configuration mode for the appropriate instance.
Example:

```
Router(config)# router isis core
```
- Step 2** Enter the **interface controller-type rack/slot/instance/port** command to configure the specific IS-IS interface.
Example:

```
Router(config-isis)# interface HundredGigE0/0/2/0
```
- Step 3** Enter the **address-family ipv4 unicast** command to enter the IPv4 address family configuration if MPLS segment routing is used.
Example:

```
Router(config-isis)# address-family ipv4 unicast
```
- Step 4** Enter the **adjacency-sid absolute adjacency-sid-value** command to configure a persistent SR adjacency SID on the interface.
Example:

```
Router(config-isis-if-af)# adjacency-sid absolute 15001
```

The adjacency SID value must be assigned from the Segment Routing Local Block (SRLB) of the router. The default SRLB range in IOS XR is 15,000 to 15,999. Configure other SR adjacency SIDs as appropriate.

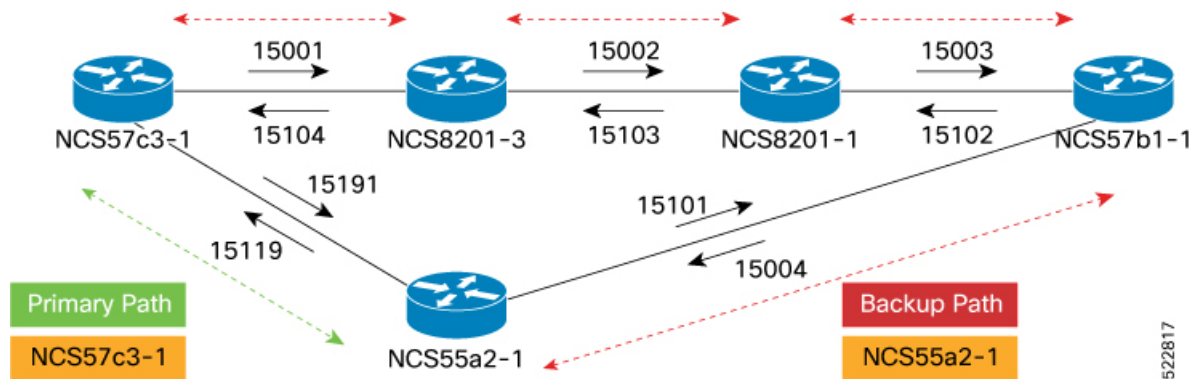
Define paths between PLE endpoints

Use this task to define network paths between PLE endpoints, enabling reliable and optimized connectivity for PLE services.



Note Use this task only with static circuit-style SR-TE policies.

Figure 2: Working and reverse path definitions



Configure the NCS57C3-1 router to set the working forward path according to the illustration. The working forward path is a single hop between directly connected routers.

Procedure

Step 1 Enter the **segment-routing** command to enter segment routing configuration mode.

Example:

```
Router(config)# segment-routing
```

Step 2 Enter the **traffic-engineering** command to enter segment routing TE configuration.

Example:

```
Router(config-sr)# traffic-engineering
```

Step 3 Enter the **segment-list** *working-forward-path* command to create segment list named *working-forward-path*.

Example:

```
Router(config-sr-te)# segment-list working-forward-path
```

Step 4 Enter the **index** command to set the index to the first hop interface adjacency SID.

Example:

```
Router(config-sr-te-sl)# index 1 mpls label 15191
```

The working and protection paths are defined for the selected PLE endpoints.

Example

Configure the NCS57C3-1 router to define all the paths between PLE endpoints according to the illustration. The working forward path is a single hop between directly connected routers. The working reverse path is a single hop between directly connected routers. The protect forward path has four interface hops between the near-end and far-end router. The protect reverse path has four interface hops between the near-end and far-end router.

```
Router(config)# segment-routing

Router(config-sr)# traffic-engineering
Router(config-sr-te)# segment-list working-forward-path
Router(config-sr-te-sl)# index 1 mpls label 15191
!
Router(config-sr-te)# segment-list working-reverse-path

Router(config-sr-te-sl)# index 1 mpls label 15119
!
Router(config-sr-te)# segment-list protect-forward-path
Router(config-sr-te-sl)# index 1 mpls label 15001
Router(config-sr-te-sl)# index 2 mpls label 15002
Router(config-sr-te-sl)# index 3 mpls label 15003
Router(config-sr-te-sl)# index 4 mpls label 15004
!
Router(config-sr-te)# segment-list protect-reversepath
Router(config-sr-te-sl)# index 1 mpls label 15101
Router(config-sr-te-sl)# index 2 mpls label 15102
Router(config-sr-te-sl)# index 3 mpls label 15103
Router(config-sr-te-sl)# index 4 mpls label 15104
```

Configure the NCS55A2-1 router to define all the paths between PLE endpoints according to the illustration.

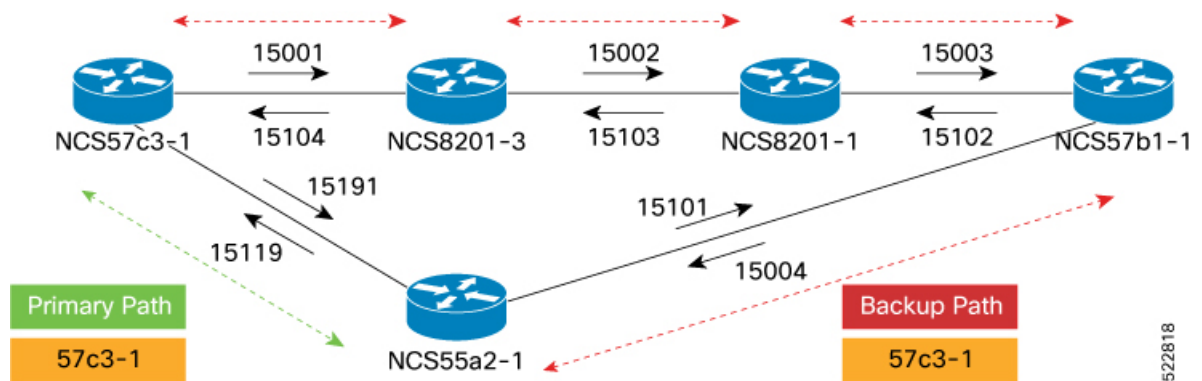
```
Router(config)# segment-routing

Router(config-sr)# traffic-engineering
Router(config-sr-te)# segment-list working-forward-path
Router(config-sr-te-sl)# index 1 mpls label 15119
!
Router(config-sr-te)# segment-list working-reverse-path
Router(config-sr-te-sl)# index 1 mpls label 15191
!
Router(config-sr-te)# segment-list protect-forward-path
Router(config-sr-te-sl)# index 1 mpls label 15101
Router(config-sr-te-sl)# index 2 mpls label 15102
Router(config-sr-te-sl)# index 3 mpls label 15103
Router(config-sr-te-sl)# index 4 mpls label 15104
!
Router(config-sr-te)# segment-list protect-reversepath
Router(config-sr-te-sl)# index 1 mpls label 15001
Router(config-sr-te-sl)# index 2 mpls label 15002
Router(config-sr-te-sl)# index 3 mpls label 15003
Router(config-sr-te-sl)# index 4 mpls label 15004
```

Configure static circuit-style SR-TE policies

Use this task to configure static circuit-style Segment Routing Traffic Engineering (SR-TE) policies, enabling predefined and predictable traffic paths within the network.

Figure 3: Segment routing traffic engineering policy



Configure the NCS57C3-1 router.

Before you begin

Static circuit-style SR-TE policies require IOS XR R7.7.1 or later software.

Procedure

Step 1 Enter the segment routing traffic engineering configuration mode.

Example:

```
Router(config)# segment-routing
Router(config-sr)# traffic-engineering
```

Step 2 Enter the **policy** *policy-name* command to create a SR-TE policy.

Example:

```
Router(config-sr-te)# policy to-55a2-1
```

Step 3 Enter the **color** *color* **end-point** **ipv4** *loopback-address* command to configure the SR-TE policy with the user-defined color and loopback address of the remote PLE far-end router.

Example:

```
Router(config-sr-te-policy)# color 1001 end-point ipv4 10.0.0.44
```

Step 4 Enter the **path-protection** command to enable path protection on the SR-TE policy.

Example:

```
Router(config-sr-te-policy)# path-protection
```

When path protection is configured, the near-end router keeps the protect path in a warm standby state. This allows it to quickly transition to the protect path if the working path is down.

Step 5 Enter the **candidate-paths** command to configure the SR working and protect candidate paths.

Example:

```
Router(config-sr-te-policy) # candidate-paths
```

Step 6 Enter the **preference preference-value** command to set the user-defined preference on the protect path.

Example:

```
Router(config-sr-te-policy-path) # preference 50
```

Paths with higher preference values are chosen over those with lower preference values.

Note

You must always set the protect path with a lower preference value.

Step 7 Enter the **explicit segment-list** command to configure the candidate path to use an explicit segment list.

Example:

```
Router(config-sr-te-policy-path-pref) # explicit segment-list protect-forward-path
```

The policy uses the previously defined *protect-forward-path* explicit segment list.

Step 8 Enter the **reverse-path segment-list** command to configure the reverse path to ensure that the SR-TE policy uses a corouted bidirectional path.

Example:

```
Router(config-sr-te-policy-path-pref) # reverse-path segment-list protect-reverse-path
```

The policy uses the previously defined *protect-reverse-path* explicit segment list.

Configure the higher-priority working path similar to the protect path using these commands:

```
Router(config-sr-te-policy-path) # preference 100
Router(config-sr-te-policy-path-pref) # explicit segment-list working-forward-path
Router(config-sr-te-policy-path-pref) # reverse-path segment-list working-reverse-path
```

Step 9 Configure a user-defined liveness policy.

Example:

```
Router(config) # performance-measurement
Router(config-perf-meas) # liveness-detection
Router(config-perf-meas) # liveness-profile name liveness-check
```

The policy uses the previously defined liveness profile named *liveness-check*.

Example

Configure the NCS55A2-1 router.

```
Router(config) # segment-routing
Router(config-sr) # traffic-eng
Router(config-sr-te) # policy to-57c3-1
Router(config-sr-te-policy) # color 1001 end-point ipv4 10.0.0.42
Router(config-sr-te-policy) # path-protection
Router(config-sr-te-policy) # candidate-paths
```

```

Router(config-sr-te-policy-path)# preference 50
Router(config-sr-te-policy-path-pref)# explicit segment-list protect-forward-path
Router(config-sr-te-policy-path-pref)# reverse-path segment-list protect-reverse-path
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list working-forward-path
Router(config-sr-te-policy-path-pref)# reverse-path segment-list working-reverse-path
Router(config)# performance-measurement
Router(config-perf-meas)# liveness-detection
Router(config-perf-meas)# liveness-profile name liveness-check

```

Configure dynamic circuit-style SR-TE policies without bandwidth CAC

Use this task to configure dynamic circuit-style Segment Routing Traffic Engineering (SR-TE) policies without enabling bandwidth Call Admission Control (CAC), allowing for flexible path selection without bandwidth reservation constraints.

Before you begin

Dynamic circuit-style SR-TE policies without bandwidth CAC require this software:

- IOS XR R7.8.1 or later
- SR-PCE 7.8.1 or later

Procedure

Step 1 Enter the segment routing traffic engineering configuration mode.

Example:

```

Router(config)# segment-routing
Router(config-sr)# traffic-engineering

```

Step 2 Enter the **policy** *policy-name* command to create a SR-TE policy.

Example:

```

Router(config-sr-te)# policy to-55a2-1

```

Step 3 Enter the **color** *color* **end-point** **ipv4** *loopback-address* command to configure the SR-TE policy with the user-defined color and loopback address of the remote PLE far-end router.

Example:

```

Router(config-sr-te-policy)# color 119 end-point ipv4 10.0.0.44

```

Step 4 Enter the **path-protection** command to enable path protection on the SR-TE policy.

Example:

```

Router(config-sr-te-policy)# path-protection

```

When path protection is configured, the near-end router keeps the protect path in a warm standby state. This allows it to quickly transition to the protect path if the working path is down.

Step 5 Enter the **candidate-paths dynamic pcep** command to configure the SR working, protect, and restoration candidate paths.

Example:

```
Router(config-sr-te-policy)# candidate-paths dynamic pcep
```

The **candidate-paths** parameter is set to **dynamic->pcep** to enable SR-PCE delegation. The dynamic computation uses standard metrics, including traffic engineering, IGP, and latency.

Step 6 Enter the **preference preference-value** command to set the user-defined preference on the protect path.

Example:

```
Router(config-sr-te-policy-path)# preference 50
```

Paths with higher preference values are chosen over those with lower preference values.

Note

You must always set the protect path with a lower preference value.

A third candidate path with a preference lower than 100 can be defined as a restoration path. This path is not pre-sigaled and must be configured with **backup-ineligible**. While the restoration path does not have a disjoint-path ID, it uses a bidirectional association ID that matches the reverse policy.

The **lock-duration** parameter determines the reversion behavior. When a higher-preference path becomes available, the policy switches to it after the lock-duration expires. By default, the lock-duration is set to 300 seconds. Setting the value to 0 disables dynamic reversion.

Step 7 Enter the **constraints segments protection** command to configure the protection behavior constraints of adjacency SID.

Example:

```
Router(config-sr-te-policy-path)# constraints segments protection {unprotected-only | adjacency-sid-only}
```

Circuit-style uses the segments constraints of **unprotected-only** and **adjacency-sid-only**. If manual adjacency-SIDs are configured, they are used; otherwise, dynamic adjacency-SIDs are used. It is recommended to configure manual adjacency-SIDs.

Step 8 Enter the **disjoint-path group-id 10 type {link | node | srlg | srlg-node}** command to configure the disjoint-path constraints.

The **group-id** for the disjoint-path must be identical on both the active and backup paths. Additionally, the **group-id** must be unique among nodes that use the same SR-PCE

The co-routed association ID must be the same for both bidirectional active and backup paths.

Step 9 Configure a user-defined liveness policy.

Example:

```
Router(config)# performance-measurement
Router(config-perf-meas)# liveness-detection
Router(config-perf-meas)# liveness-profile name liveness-check
```

The policy uses the previously defined liveness profile named *liveness-check*.

This example configures the dynamic circuit-style SR-TE policy without bandwidth CAC.

```
policy dynamic-cs-srte-to-55a2-p2
  color 119 end-point ipv4 10.0.0.44
```

```

path-protection
!
candidate-paths
preference 100
dynamic
  pcep
  !
  metric
  type igp
  !
  !
constraints
  segments
  protection unprotected-only
  adjacency-sid-only
  !
  disjoint-path group-id 10 type link
  !
bidirectional
  co-routed
  association-id 101
  !
!
preference 200
dynamic
  pcep
  !
  metric
  type igp
  !
  !
lock
  duration 30
  !
constraints
  segments
  protection unprotected-only
  adjacency-sid-only
  !
  disjoint-path group-id 10 type link
  !
bidirectional
  co-routed
  association-id 201
  !
!
preference 50
dynamic
  pcep
  !
  metric
  type igp
  !
  !
backup-ineligible
lock
  duration 60
  !
constraints
  segments
  protection unprotected-only
  adjacency-sid-only
  !
  !

```

```

bidirectional
  co-routed
  association-id 301
  !
!
!
performance-measurement
  liveness-detection
  liveness-profile backup name CS-PROTECT
  liveness-profile name CS-WORKING
  invalidation-action down
  
```

Configure dynamic circuit-style SR-TE policies with bandwidth CAC

Use this task to configure dynamic circuit-style SR-TE policies with bandwidth Call Admission Control (CAC).

Dynamic PLE with bandwidth CAC is supported through Crosswork Network Controller. The bandwidth constraint is specified for the circuit-style policy. The Cisco Crosswork Optimization Engine application is included in Circuit Style Manager.

The circuit-style Segment Routing Traffic Engineering (CS SR-TE) feature pack provides a bandwidth-aware Path Computation Element (PCE) to compute CS SR-TE policy paths that you can visualize in your network. CS SR-TE policies guarantee allocated bandwidth services with predictable latency and persistent bidirectional path protection of critical traffic. CS SR-TE reserves a percentage of bandwidth in the network and computes bidirectional failover paths for CS SR-TE policies using the requested bandwidth.

Circuit Style Manager requires you to configure policies on each head-end node using matching association IDs, bandwidth values, constraints, computation type, and source and endpoint IP addresses.

Figure 4: Crosswork Optimization Engine—SR-PCE communication flow diagram

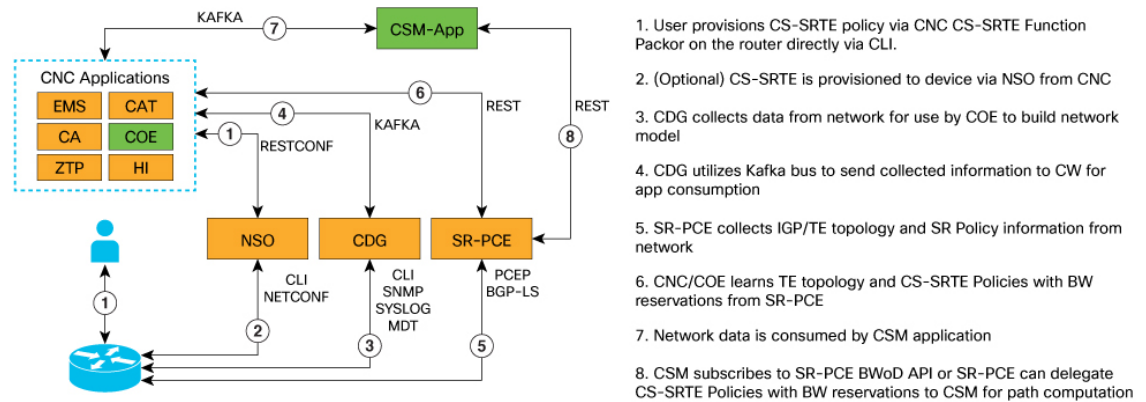
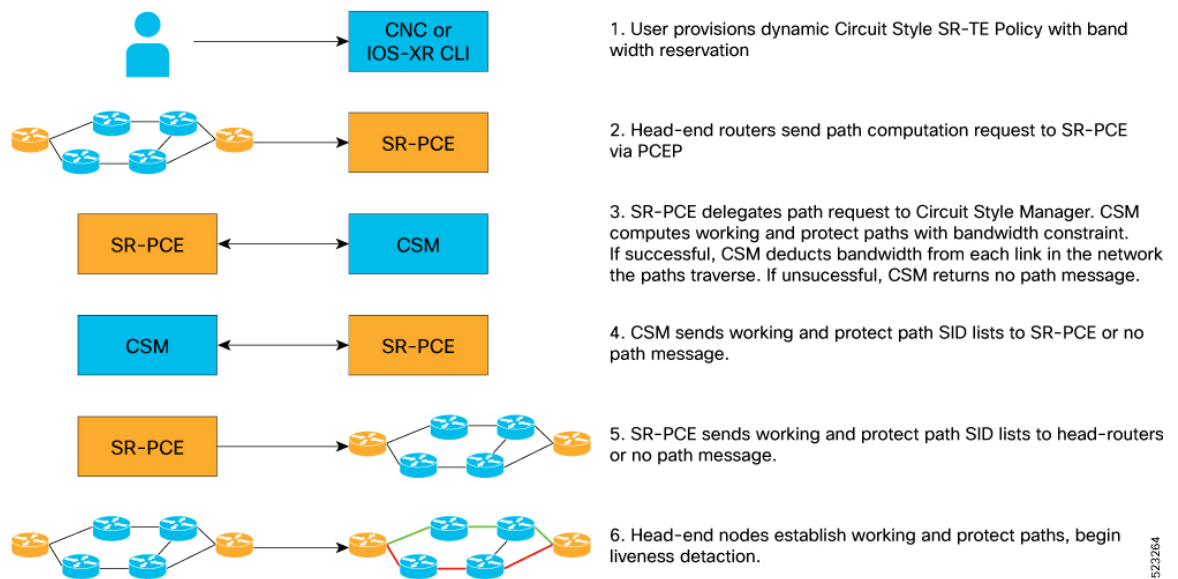


Figure 5: Crosswork Optimization Engine —SR-PCE call flow diagram



523264

Before you begin

Dynamic circuit-style SR-TE policies with bandwidth CAC require this software:

- IOS XR R7.9.1 or later
- SR-PCE 7.9.1 or later
- Crosswork Optimization Engine 5.0 or later

Procedure

-
- Step 1** Enable the circuit-style Segment Routing Traffic Engineering (CS SR-TE) feature pack.
- Step 2** Configure CS SR-TE policies on the devices.
- Step 3** Verify that the CS SR-TE policies appear in the SR Policy table.
-

Enable circuit-style SR-TE using Crosswork Network Controller

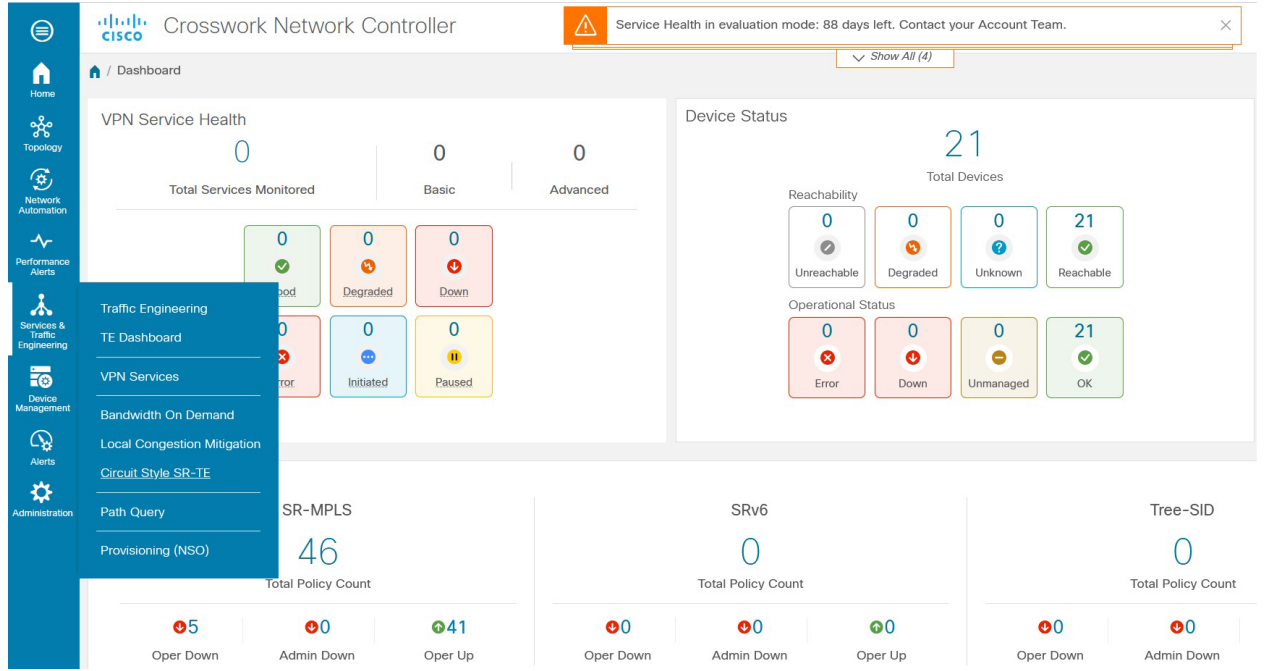
Use this task to enable circuit-style Segment Routing Traffic Engineering (CS SR-TE) policies using Cisco Crosswork Network Controller.

To manage and visualize CS-SR policies on the topology map, you must first enable CS SR-TE and set the bandwidth reservation settings.

When CS SR-TE is enabled, it computes the best failover bidirectional paths between the two nodes, using the requested bandwidth and other constraints defined in the CS SR-TE policy configuration.

Procedure

Step 1 From the main menu, choose **Services & Traffic Engineering > Circuit Style SR-TE > Configuration** .



Step 2 Toggle the **Enable** switch to **True**.

Step 3 Enter the required bandwidth pool size and the threshold information.

523258

523259

| Option | Description |
|---------------------------------|--|
| Basic Pane | |
| Link CS BW Pool Size | The percentage of bandwidth that each link reserves for CS-SR policies. |
| Link CS BW Min Threshold | An alert is generated when the bandwidth utilization percentage on a link used for a CS-SR policy exceeds this value. |
| Advanced Pane | |
| Validation Interval | The interval that CS-SR policy will wait before the bandwidth that is reserved for an undelegated policy is returned to the CS-SR policy bandwidth pool. |
| Timeout | The duration until which CS-SR policy will wait for the delegation request, to generate a notification. |
| Restore Delegation Delay | The duration until which CS-SR policy will pause before processing a restore path delegation. |

Step 4 Click **Commit Changes** to save the configuration.

View circuit-style SR-TE policy details using Crosswork Network Controller

Use this task to view details about circuit-style Segment Routing Traffic Engineering (CS SR-TE) policies using Cisco Crosswork Network Controller.

Procedure

Step 1 From the main menu, choose **Services & Traffic Engineering > SR-MPLS** and click **Circuit Style**.

The **SR Policy** table lists all the CS SR-TE policies.

Step 2 From the **Actions** column, click **View Details** for one of the CS SR-TE policies.

The **Circuit Style Policy Details** window is displayed in the side panel. By default, the active path is displayed and shows the bidirectional paths (**Bi-Dir Path** checkbox is selected) on the topology map.

The screenshot shows the Cisco Crosswork Network Controller interface. The main area displays a map of the United States with network nodes and paths highlighted. The details panel on the right shows the following information:

- Current** tab selected.
- Headend:** ron-ncs55a2-1 | TE RID: 100.0.0.44 PCC IP: 100.0.0.44 Source IP: 100.0.0.44
- Endpoint:** ron-ncs57c3-1 | TE RID: 100.0.0.42 Dest IP: 100.0.0.42 Color: 119
- Summary:**
 - Admin State: Up
 - Oper State: Up
 - Binding SID: 27462
 - Policy Type: Circuit-Style
 - Profile ID: -
 - Description: -
 - Traffic Rate: 0 Mbps
 - Unused: True
- Candidate Path:**

| Path Name | Pref | State |
|------------------------------------|------|-------|
| > cfg_dynamic-cs-srte-to-57c3-p... | 200 | Up |
| > cfg_dynamic-cs-srte-to-57c3-p... | 100 | Up |

If the bandwidth of a new policy exceeds bandwidth pool size, the policy goes down. The requested bandwidth is set to the configured bandwidth, but the reserved bandwidth becomes zero.

If an existing policy is configured with a bandwidth that exceeds the bandwidth pool size, the policy remains up, but the reserved bandwidth stays at its previous value.

The sample output displays a policy's requested bandwidth and current bandwidth.

```
RP0/RP0/CPU0:ron-ncs57c3-1#show segment-routing traffic-eng policy color 119
Thu May 4 09:01:57.797 PDT
```

SR-TE policy database

Color: 119, End-point: 10.0.0.44
Name: srte_c_119_ep_10.0.0.44
Status:
Admin: up Operational: up for 11:46:42 (since May 3 21:15:14.898)
Candidate-paths:
Preference: 200 (configuration) (active)
Name: dynamic-cs-srte-to-55a2-p2
Requested BSID: dynamic
PCC info:
Symbolic name: cfg_dynamic-cs-srte-to-55a2-p2_discr_200
PLSP-ID: 7
Constraints:
Protection Type: unprotected-only
Maximum SID Depth: 12
Adjacency SIDs Only: True
Dynamic (pce 201.0.0.106) (valid)
Metric Type: IGP, Path Accumulated Metric: 20
SID[0]: 15739 [Adjacency-SID, 10.139.1.1 - 10.139.1.2]
SID[1]: 15720 [Adjacency-SID, 10.1.26.0 - 10.1.26.1]
Reverse path:
SID[0]: 15227 [Adjacency-SID, 10.1.26.1 - 10.1.26.0]
SID[1]: 15727 [Adjacency-SID, 10.139.1.2 - 10.139.1.1]
Protection Information:
Role: WORKING
Path Lock: Timed
Lock Duration: 300(s)
State: ACTIVE
Preference: 100 (configuration) (protect)
Name: dynamic-cs-srte-to-55a2-p2
Requested BSID: dynamic
PCC info:
Symbolic name: cfg_dynamic-cs-srte-to-55a2-p2_discr_100
PLSP-ID: 6
Constraints:
Protection Type: unprotected-only
Maximum SID Depth: 12
Adjacency SIDs Only: True
Dynamic (pce 201.0.0.106) (valid)
Metric Type: IGP, Path Accumulated Metric: 30
SID[0]: 15734 [Adjacency-SID, 10.1.42.0 - 10.1.42.1]
Reverse path:
SID[0]: 15523 [Adjacency-SID, 10.1.42.1 - 10.1.42.0]
Protection Information:
Role: PROTECT
Path Lock: Timed
Lock Duration: 300(s)
State: STANDBY
Attributes:
Binding SID: 24047
Forward Class: Not Configured
Steering labeled-services disabled: no
Steering BGP disabled: no
IPv6 caps enable: yes
Bandwidth Requested: 35.000 Gbps
Bandwidth Current: 35.000 Gbps
Invalidation drop enabled: no
Max Install Standby Candidate Paths: 0

Configure PLE over EVPN-VPWS

Use this task to configure PLE over Ethernet VPN Virtual Private Wire Service (EVPN-VPWS), enabling the delivery of emulated private line services.

The two core components are the Pseudowire class, which specifies the transport type, and the xconnect service, which configures the EVPN-VPWS service parameters.

Figure 6: PLE EVPN-VPWS service



Configure the NCS57C3-1 router.

Procedure

Step 1 Enter the **l2vpn** command to enter l2vpn configuration mode.

Example:

```
Router(config)# l2vpn
```

Step 2 Enter the **pw-class user-defined-name** command to define the Pseudowire class used with the PLE service.

Example:

```
Router(config-l2vpn)# pw-class circuit-style-srte
```

The same Pseudowire class can be used for multiple PLE services between the same far-end routers, or a unique class can be used for each service. The Pseudowire class defines the underlying transport for the service. In this case, the transport is MPLS, using a specific SR-TE policy. The **pw-class** command is followed by a user-defined name. In this example, the name is *circuit-style-srte*.

Step 3 Enter the **encapsulation mpls** command to configure transport encapsulation.

Example:

```
Router(config-l2vpn-pwc)# encapsulation mpls
```

Step 4 Enter the **preferred-path sr-te policy** command to configure an explicit MPLS path to be used for the L2VPN service.

Example:

```
Router(config-l2vpn-pwc-mpls)# preferred-path sr-te policy srte_c_1001_ep_10.0.0.42
```

The preferred-path is the circuit-style policy used to configure an explicit MPLS path for the L2VPN service. In this case, the preferred-path is set to the SR-TE policy created from the source node to the 10.0.0.42 endpoint node. The path refers to the computed name of the policy, not the configured name.

Step 5 Enter the **xconnect group ple** command to enter xconnect configuration mode.

Example:

```
Router(config-l2vpn)# xconnect group ple
```

The **xconnect group** command is used to administratively group similar L2VPN services and can be set to any user-defined value.

Step 6 Enter the **p2p user-defined-name** command to create point-to-point L2VPN service with user-defined name *ple-cs-1*.

Example:

```
Router(config-l2vpn-xc) # p2p ple-cs-1
```

Step 7 Enter the **interface cem** command to specify the client interface.

Example:

```
Router(config-l2vpn-xc-p2p) # interface cem 0/0/2/1
```

This interface is the CEM interface type for PLE.

Step 8 Enter the **neighbor evpn** command to configure EVPN-VPWS parameters.

Example:

```
Router(config-l2vpn-xc-p2p) # neighbor evpn evi 100 target 4201 source 4401
```

The user-defined *evi* value must be the same on each far-end router. The target must match the source value configured on the remote endpoint.

Step 9 Enter the **pw-class** command to attach the previously defined Pseudowire class *circuit-style-te* to the L2VPN service.

Example:

```
Router(config-l2vpn-xc-p2p-pw) # pw-class circuit-style-srte
```

Configure the NCS55a2-1 router.

```
Router(config) # l2vpn
Router(config-l2vpn) # pw-class circuit-style-srte
Router(config-l2vpn-pwc) # encapsulation mpls
Router(config-l2vpn-pwc-mpls) # preferred-path sr-te policy srte_c_1001_ep_10.0.0.44
Router(config-l2vpn) # xconnect group ple
Router(config-l2vpn-xc) # p2p ple-cs-1
Router(config-l2vpn-xc-p2p) # interface cem0/0/2/1
Router(config-l2vpn-xc-p2p) # neighbor evpn evi 100 target 4201 source 4401
Router(config-l2vpn-xc-p2p-pw) # pw-class circuit-style-srte
```

Verify PLE configurations

Verify PLE configurations using these commands.

Procedure

Step 1 Use this command to show the general CEM statistics.

Example:

```

RP/0/RP0/CPU0:ron-ncs57c3-1#show controllers CEM 0/0/3/1

Sat Sep 24 11:34:22.533 PDT
Interface : CEM0/0/3/1
Admin state : Up
Oper state : Up
Port bandwidth : 10312500 kbps
Dejitter buffer (cfg/oper/in-use) : 0/813/3432 usec
Payload size (cfg/oper) : 1280/1024 bytes
PDV (min/max/avg) : 980/2710/1845 usec
Dummy mode : last-frame
Dummy pattern : 0xaa
Idle pattern : 0xff
Signalling : No CAS
RTP : Enabled
Clock type : Differential
Detected Alarms : None
Statistics Info
-----
Ingress packets : 517617426962, Ingress packets drop : 0
Egress packets : 517277124278, Egress packets drop : 0
Total error : 0
Missing packets : 0, Malformed packets : 0
Jitter buffer underrun : 0, Jitter buffer overrun : 0
Misorder drops : 0
Reordered packets : 0, Frames fragmented : 0
Error seconds : 0, Severely error seconds : 0
Unavailable seconds : 0, Failure counts : 0
Generated L bits : 0, Received L bits : 0
Generated R bits : 339885178, Received R bits : 17
Endpoint Info
-----
Passthrough : No

```

Step 2 Use this command to show the PM statistics for 30 seconds.

Example:

```

RP/0/RP0/CPU0:ron-ncs57c3-1#show controllers CEM 0/0/3/1 pm current 30-sec cem

Sat Sep 24 11:37:02.374 PDT
CEM in the current interval [11:37:00 - 11:37:02 Sat Sep 24 2022]
CEM current bucket type : Valid
INGRESS-PKTS : 2521591 Threshold : 0 TCA(enable) : NO
EGRESS-PKTS : 2521595 Threshold : 0 TCA(enable) : NO
INGRESS-PKTS-DROPPED : 0 Threshold : 0 TCA(enable) : NO
EGRESS-PKTS-DROPPED : 0 Threshold : 0 TCA(enable) : NO
INPUT-ERRORS : 0 Threshold : 0 TCA(enable) : NO
OUTPUT-ERRORS : 0 Threshold : 0 TCA(enable) : NO
MISSING-PKTS : 0 Threshold : 0 TCA(enable) : NO
PKTS-REORDER : 0 Threshold : 0 TCA(enable) : NO
JTR-BFR-UNDERRUNS : 0 Threshold : 0 TCA(enable) : NO
JTR-BFR-OVERRUNS : 0 Threshold : 0 TCA(enable) : NO
MIS-ORDER-DROPPED : 0 Threshold : 0 TCA(enable) : NO
MALFORMED-PKT : 0 Threshold : 0 TCA(enable) : NO
ES : 0 Threshold : 0 TCA(enable) : NO
SES : 0 Threshold : 0 TCA(enable) : NO
UAS : 0 Threshold : 0 TCA(enable) : NO
FC : 0 Threshold : 0 TCA(enable) : NO
TX-LBITS : 0 Threshold : 0 TCA(enable) : NO
TX-RBITS : 0 Threshold : 0 TCA(enable) : NO
RX-LBITS : 0 Threshold : 0 TCA(enable) : NO
RX-RBITS : 0 Threshold : 0 TCA(enable) : NO

```

Step 3 Use this command to show the controller information for Ethernet 10GE.

Example:

```
RP/0/RP0/CPU0:ron-ncs57c3-1#show controllers TenGigECtrlr0/0/3/2

Sat Sep 24 11:43:23.164 PDT
Operational data for interface TenGigECtrlr0/0/3/2:
State:
Administrative state: enabled
Operational state: Up
LED state: Green On
PRBS:
Status: Locked
Mode: Source-sink
Pattern: User-defined
Direction: System
Error-inject: None
Framing: Unframed
User-pattern: 0xabcd0123456789
Phy:
Media type: Not known
Autonegotiation disabled.
Operational values:
Speed: 10Gbps
Duplex: Full Duplex
Flowcontrol: None
Loopback: None (or external)
Inter-packet gap: standard (12)
BER monitoring:
Not supported
```

Step 4 Use this command to show the PM statistics for the Ethernet controller.

Example:

```
RP/0/RP0/CPU0:ron-ncs57c3-1#show controllers TenGigECtrlr0/0/3/2 pm current 30-sec ether

Sat Sep 24 11:45:39.919 PDT
ETHER in the current interval [11:45:30 - 11:45:39 Sat Sep 24 2022]
ETHER current bucket type : Valid
RX-PKT : 4528985 Threshold : 0 TCA(enable) : NO
STAT-PKT : 9057971 Threshold : 0 TCA(enable) : NO
OCTET-STAT : 2318840576 Threshold : 0 TCA(enable) : NO
OVERSIZE-PKT : 0 Threshold : 0 TCA(enable) : NO
FCS-ERR : 0 Threshold : 0 TCA(enable) : NO
LONG-FRAME : 0 Threshold : 0 TCA(enable) : NO
JABBER-STATS : 0 Threshold : 0 TCA(enable) : NO
64-OCTET : 0 Threshold : 0 TCA(enable) : NO
65-127-OCTET : 0 Threshold : 0 TCA(enable) : N
128-255-OCTET : 0 Threshold : 0 TCA(enable) : NO
256-511-OCTET : 9057971 Threshold : 0 TCA(enable) : NO
512-1023-OCTET : 0 Threshold : 0 TCA(enable) : NO
1024-1518-OCTET : 0 Threshold : 0 TCA(enable) : NO
IN-UCAST : 0 Threshold : 0 TCA(enable) : NO
IN-MCAST : 0 Threshold : 0 TCA(enable) : NO
IN-BCAST : 0 Threshold : 0 TCA(enable) : NO
OUT-UCAST : 0 Threshold : 0 TCA(enable) : NO
OUT-BCAST : 0 Threshold : 0 TCA(enable) : NO
OUT-MCAST : 0 Threshold : 0 TCA(enable) : NO
TX-PKT : 4528986 Threshold : 0 TCA(enable) : NO
OUT-OCTET : 1159420416 Threshold : 0 TCA(enable) : NO
IFIN-ERRORS : 0 Threshold : 0 TCA(enable) : NO
IFIN-OCTETS : 0 Threshold : 0 TCA(enable) : NO
STAT-MULTICAST-PKT : 0 Threshold : 0 TCA(enable) : NO
```

```

STAT-BROADCAST-PKT : 0 Threshold : 0 TCA(enable) : NO
STAT-UNDERSIZED-PKT : 0 Threshold : 0 TCA(enable) : NO
IN_GOOD_BYTES : 1159420160 Threshold : 0 TCA(enable) : NO
IN_802_1Q_FRAMES : 0 Threshold : 0 TCA(enable) : NO
IN_GOOD_PKTS : 4528985 Threshold : 0 TCA(enable) : NO
IN_DROP_OTHER : 0 Threshold : 0 TCA(enable) : NO
OUT_GOOD_BYTES : 1159420416 Threshold : 0 TCA(enable) : No
OUT_802_1Q_FRAMES : 0 Threshold : 0 TCA(enable) : NO
OUT_GOOD_PKTS : 4528986 Threshold : 0 TCA(enable) : NO
IN_ERROR_FRAGMENTS : 0 Threshold : 0 TCA(enable) : NO
IN_PKT_64_OCTET : 0 Threshold : 0 TCA(enable) : NO
IN_PKTS_65_127_OCTETS : 0 Threshold : 0 TCA(enable) : NO
IN_PKTS_128_255_OCTETS : 0 Threshold : 0 TCA(enable) : NO
IN_PKTS_256_511_OCTETS : 4528985 Threshold : 0 TCA(enable) : NO
IN_PKTS_512_1023_OCTETS : 0 Threshold : 0 TCA(enable) : NO
IN_PKTS_1024_1518_OCTETS : 0 Threshold : 0 TCA(enable) : NO
OUT_PKT_64_OCTET : 0 Threshold : 0 TCA(enable) : NO
OUT_PKTS_65_127_OCTETS : 0 Threshold : 0 TCA(enable) : NO
OUT_PKTS_128_255_OCTETS : 0 Threshold : 0 TCA(enable) : NO
OUT_PKTS_256_511_OCTETS : 4528986 Threshold : 0 TCA(enable) : NO
OUT_PKTS_512_1023_OCTETS : 0 Threshold : 0 TCA(enable) : NO
OUT_PKTS_1024_1518_OCTETS : 0 Threshold : 0 TCA(enable) : NO
TX_UNDERSIZED_PKT : 0 Threshold : 0 TCA(enable) : NO
TX_OVERSIZED_PKT : 0 Threshold : 0 TCA(enable) : NO
TX_FRAGMENTS : 0 Threshold : 0 TCA(enable) : NO
TX_JABBER : 0 Threshold : 0 TCA(enable) : NO
TX_BAD_FCS : 0 Threshold : 0 TCA(enable) : NO

```

Last clearing of "show controllers ETHERNET" counters never

Step 5 Use this command to show the controller information for Fiber Channel 8G.

Example:

```
RP/0/RP0/CPU0:ron-ncs57c3-1#show controllers EightGigFibreChanCtrlr0/0/3/4
```

```

Sat Sep 24 11:53:09.820 PDT
Operational data for Fibre Channel controller EightGigFibreChanCtrlr0/0/3/4
State:
Admin State : Up
Operational state : Up
LED state : Green On
Secondary admin state : Normal
Laser Squelch : Disabled
Performance Monitoring is enabled
Operational values:
Speed : 8 Gbps
Loopback : None
BER monitoring:
Signal Degrade : 1e-0
Signal Fail : 1e-0
Hold-off Time : 0 ms
Forward Error Correction : Not Configured

```

Step 6 Use this command to show the PM statistics for Fiber Channel.

Example:

```

RP/0/RP0/CPU0:ron-ncs57c3-1#show controllers EightGigFibreChanCtrlr0/0/3/4 pm current 30-sec fc
Sat Sep 24 11:51:55.168 PDT
FC in the current interval [11:51:30 - 11:51:55 Sat Sep 24 2022]
FC current bucket type : Valid
IFIN-OCTETS : 16527749196 Threshold : 0 TCA(enable) : NO
RX-PKT : 196758919 Threshold : 0 TCA(enable) : NO
IFIN-ERRORS : 0 Threshold : 0 TCA(enable) : NO

```

```
RX-BAD-FCS : 0 Threshold : 0 TCA(enable) : NO
IFOUT-OCTETS : 0 Threshold : 0 TCA(enable) : NO
TX-PKT : 0 Threshold : 0 TCA(enable) : NO
TX-BAD-FCS : 0 Threshold : 0 TCA(enable) : NO
RX-FRAMES-TOO-LONG : 0 Threshold : 0 TCA(enable) : NO
RX-FRAMES-TRUNC : 0 Threshold : 0 TCA(enable) : NO
TX-FRAMES-TOO-LONG : 0 Threshold : 0 TCA(enable) : NO
TX-FRAMES-TRUNC : 0 Threshold : 0 TCA(enable) : NO
```
