# Cisco Crosswork Network Controller 7.0 Solution Workflow Guide

**First Published:** 2024-07-22

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

# CONTENTS

CHAPTER 1

# Solution Overview

This section explains the following topics:



## Description

The exponential growth of network traffic and the pressures of efficiently running network operations pose huge challenges for network operators. Providing quick, intent-based service delivery and optimal network utilization, with the ability to react to bandwidth and latency demand fluctuations in real-time is vital to success. Migration to Software-Defined Networks (SDNs) and automation of operational tasks is the optimal way for operators to accomplish these goals.

Cisco Crosswork Network Controller is an integrated network automation solution for deploying and operating IP transport networks that delivers increased service agility, cost efficiency, and optimization for faster time-to-customer value and lower operating costs. The solution combines intent-based network automation to deliver critical capabilities for service orchestration and fulfillment, network optimization, service path computation, device deployment and management, and anomaly detection with operator-selected manual or automated remediation. Cisco Crosswork Network Controller delivers network optimization capabilities that are nearly impossible to replicate even with a highly skilled and dedicated staff operating the network.

The fully integrated solution combines functionality from multiple Crosswork components installed upon a common Crosswork infrastructure, as well as industry-leading capabilities from Cisco® Network Services Orchestrator (NSO), Cisco Segment Routing Path Computation Element (SR-PCE), and Cisco Crosswork Planning. Its unified user interface provides a single pane of glass for real-time visualization of the network topology and services, provisioning, monitoring, and optimization.

## What's New in This Release

The information below lists the primary new features and functionality introduced in Cisco Crosswork Network Controller 7.0.x.

**Platform Infrastructure**

- **AWS EC2 Support**: Support is available for deploying the following Cisco Crosswork Network Controller packages on the AWS EC2 platform.
    - Cisco Crosswork Network Controller Essentials package, excluding Zero Touch Provisioning
    - Cisco Crosswork Network Controller Advantage package, excluding Service Health
    - Cisco Crosswork Network Controller Add-on package: data export to 3rd party apps (CDG) & geo redundancy (new)

  New capabilities supported for AWS EC2 platform:
    - Element management enhancements
        - Software Image Management (SWIM) UI
        - Performance trends
        - Deeper inventory
        - Policy and service performance trend analysis
    - FlexAlgo aware BWoD
    - Support for RON 3.0 Optics
- **Crosswork Network Controller deployed as a single VM**: This release introduces support for deploying the Crosswork Network Controller solution on a single VM. The solution is deployed using a unified package that includes Crosswork Infrastructure, Embedded Collectors, and Element Management Functions, enabling you to leverage the device lifecycle functionalities of Crosswork. However, functionalities like service provisioning and overlay are only available on the cluster-based installation.
- **Geo Redundancy**: This release introduces the asynchronous data replication for geo redundancy clusters (on-premises L2 & L3 multi-site) for Crosswork Network Controller. It is no longer necessary to move the cluster into maintenance mode for data synchronization. For more information, see the *Enable Geo Redundancy* section in the Cisco Crosswork Network Controller 7.0 Installation Guide.
- **Dual stack support**: Support is available for deploying the Crosswork Network Controller with a dual stack (IPv4 and IPv6) configuration.
- **Installation enhancements**:
    - The auto-action feature has been added to automate application installation alongside cluster installation.
    - Support has been added to install Cisco Crosswork Network Controller 7.0 on VMware vCenter and ESXi version 8.0.
- **System access and security infrastructure**:
    - Support has been added to register Crosswork components using the Crosswork Network Controller solution license.
    - Support has been added to launch the Single Sign On (SSO) service provider page from the Crosswork UI.
    - Support has been added for the automatic renewal of internal Crosswork certificates.

### Traffic Engineering

- **Bandwidth on Demand (BWoD) feature pack**: You now have the option to have BWoD find a path with a specified Flexible Algorithm SID. The acceptable SID values are 0, 1, and 128-255.
- **Alarms and Events**: Traffic engineering alarms and events have been added or updated to be more consistent with other Crosswork services.
- **Virtual Routing and Forwarding (VRF)**: Duplicate IP addresses on two interfaces in the same router are now supported when configured in a VRF table.
- **Interface Index (IfIndex)**: Crosswork Network Controller now supports multiple IP addresses on a single IfIndex.
- **IS-IS Layer 1 and Layer 2**: Crosswork Network Controller now discovers L1 and L2 links. They are displayed on the topology map as dotted lines between devices.
- **Cisco WAN Automation Engine (WAE) and Cisco Crosswork Planning plan file**: The plan file from Crosswork Network Controller now includes additional attributes: LSP MetricType, Dynamic, and applicable disjoint group information. A plan file is comprised of a series of tables that store information about a network, including topology, configuration information, traffic, failure state, and visual layout. For more information, see the Cisco Crosswork Solution Workflow Guide.
- **Interface Names**: When Element Management Functions (EMF) is installed, Crosswork Network Controller now abstracts any non-standardized interface name and populates the `interfaceName` value. As a result, only the `interfaceName` is used, which helps alleviate device telemetry, polling, and configuration problems.

  The Link Summary details page displays the following field changes:

  - Interface Name—Displays the `interfaceName` value.
  - Interface Description—If applicable, displays any user specified text on the actual interface.

### Topology

- **Enhanced topology visualization of large VPNs**: Cisco Crosswork Network Controller 7.0 introduces enhanced navigation, provisioning, and visualization of the service overlay and details for large L3 VPNs containing up to 20,000 endpoints (UNI/PE-CE interface).

  When a user selects a VPN service in the UI that is too large to display in full (since a maximum of 50 endpoints can only be displayed once), they can click **Select endpoints** and choose from a list of endpoints to visualize the service overlay and details. The list shows only the endpoints on devices in the current selected device group. The list also includes filters to narrow down the list of endpoints, making it easier to select.

  Preconditions and limitations:

  - Visualization of large VPN support for L3VPN aligning with IETF L3 NM model.
  - Maximum number of vpn-node (PE) in the L3VPN: 4,000.
  - Maximum number of endpoints (UNI/PE-CE interface) in the L3VPN: 20,000.
  - Maximum number of policies or tunnels per large VPN: 5,000.
  - Maximum number of large VPN service instances in each deployment: 6.
  - Recommended provisioning of endpoints in a single request with a single L3VPN: 500.

- **Enhancements in the Topology UI - Links Visualization**: The Topology UI has been updated with a new **Links** tab to display all links on the map and a global links table in the **Devices** tab showing link details and metrics. Key metrics like bandwidth utilization, packet errors, packet drops, delay, and jitter are now visualized in both the map and details panels, with delay and jitter available when Crosswork Service Health is installed and SR-PM is enabled. You can also customize link color and metric thresholds in the Topology map and view historical data for collected metrics on the **Link Details** page.
- **Topology Dashboard**: A new **Topology** dashlet has been added to the **Dashboards** page, offering details on L2 and L3 links along with their associated metrics. When you click the L2 or L3 links in the dashboard, you will be directed to the Topology UI, where the corresponding map is displayed in the left pane. The **Devices** and **Links** tabs in the right pane offer detailed information about the devices and links on the map.

**Change Automation**

- **Support for check-sync action play**: Crosswork Network Controller 7.0 includes a new stock play, **Perform Check Sync on the device**, to achieve check-sync. You can use this Play as a pre-step to running other operations in the Playbook or as part of pre-maintenance. This Play checks the device sync status with NSO and performs a sync-from (pulling the present device config into NSO) only when needed, based on the Playbook's sync parameter value. It reduces the playbook execution time and ensures the NSO configuration matches the device configuration.
    - If the Playbook's sync parameter is set to True and the device is not in sync, it will sync the device with the NSO configuration, and the operation succeeds with an in-sync status.
    - If the sync parameter is set to False and the device is not in sync, the Playbook fails with a commit message.
    - If the device is already in sync, the operation succeeds.

**Service Health**

- **Monitor Service Health using Cisco Provider Connectivity Assurance**: Crosswork Network Controller can leverage external probes from Cisco Provider Connectivity Assurance (formerly Accedian Skylight) to provide additional insights into the health of the L3 VPN services in the network.

**Note** Cisco Provider Connectivity Assurance integration is available as a limited-availability feature in this release. Engage with your account team for more information. For more information, see the *Monitor Service Health* section in Cisco Crosswork Network Controller 7.0 Service Health Monitoring.

- **L3 VPN service monitoring enhancements**: Service Health supports large-scale VPN visualization by monitoring L3VPN services at the node level and creating an Assurance graph for each service at either the node or endpoint level. If the graph contains more than 50 endpoints, Service Health indicates that the graph is too large to view and prompts you to use the **Select Endpoints** option to select and view up to 50 endpoints.
- **Enhanced metrics and insights with SR-PM**: When Segment Routing Performance Measurement (SR-PM) is enabled on your devices, Service Health collects and processes additional metrics like Delay, Delay Variance, and Liveness to assess the performance of links and the health of TE policies. It also offers historical data and trends for these metrics, providing valuable insights into network performance and trends.

- **Service Health dashboard**: A new Service Health Dashboard displays a consolidated view of L2 VPN and L3 VPN services. In the event an SLA for a service is breached, the UI clearly indicates the break, making detection of problems easier.

**Data Gateway**

- **Support for dual-stack configurations**: Crosswork Network Controller introduces support for dual-stack configurations, enabling the system to establish connections using IPv4 and IPv6 protocols. With this enhancement, Crosswork can seamlessly communicate concurrently with various systems (such as NTP, DNS, and Syslog) and devices (SSH, SNMP, MDT) over IPv4 and IPv6. In dual-stack mode, Crosswork gives priority to IPv6 for all communication purposes.

  For information on configuring a dual stack when creating or editing a pool and adding destinations, see Cisco Crosswork Network Controller 7.0 Administration Guide.

- **New custom package to support different file formats**: Crosswork Network Controller has introduced a new feature that provides the flexibility in managing custom packages. This feature unifies the previously available different upload structures by standardizing the file structure for both system and custom packages. Crosswork Network Controller now supports multiple custom packages and allowing users to upload their specific packages more efficiently. The feature includes support for aggregate custom packages, which users can use for Embedded Collectors and Crosswork Data Gateway in a cluster deployment.

  The updated Crosswork Network Controller UI enables users to upload a common package type as well as the new aggregate package type, facilitating the combination and merging of various file formats into a single, unified package.

  For information on adding and downloading aggregate packages through the Crosswork UI, see Cisco Crosswork Network Controller 7.0 Administration Guide.

- **Deployable on VMware vCenter version 8.0**: Cisco Crosswork Network Controller 7.0 and Crosswork Data Gateway instances can be installed on VMware vCenter and ESXi version 8.0.

  For information on the installation of Crosswork Data Gateway on vCenter, see Cisco Crosswork Network Controller 7.0 Installation Guide.

- **Embedded Collectors**: With an intent to simplify deployment, the Crosswork Network Controller can be set up on a single VM, though this comes with a trade-off in terms of scale and availability. This deployment model minimizes the reliance on external components by incorporating an embedded collector, replacing the need for external Crosswork Data Gateway VMs. In this model, the data gateway is installed as a lightweight CAPP within the single VM, reducing the need for separate data gateway nodes and significantly decreasing the deployment footprint. The data gateway functions as embedded collectors within the Kubernetes pods.

  For information on installing embedded collectors, see Cisco Crosswork Network Controller 7.0 Installation Guide.

- **A new Interactive Console menu option to modify the controller's IP or FQDN for data gateway enrollment and geo redundancy features**: The interactive menu now has a new option that enables you to modify the controller's IP or FQDN in these scenarios:

  - A data gateway may fail to enroll with the Crosswork Network Controller if deployed with an invalid controller IP.

  - A data gateway is registered with a Crosswork Network Controller, and the controller's VIP IP or IP is changed to an FQDN. This change might be necessary for Geo Redundancy configuration.

For more information on using the new menu option, see the *Configure Controller IP for Crosswork Data Gateway* section in Cisco Crosswork Network Controller 7.0 Administration Guide.

**Device Lifecycle Management**

- Device management has been enhanced with new features allowing for customized monitoring and management of network devices. These include:
  - Tag Management window to manage the tags available for assignment to devices in your network. Tags can provide information such as the device's physical location and administrator's email ID, which can be used to group devices.
  - A comprehensive Network Inventory overview listing device names, types, hardware details, and operational statuses.
  - Manual inventory synchronization for up-to-date network device tracking.
  - Options for device groups and port groups are available, which can be utilized for performance monitoring data collection based on specific parameters.
- **Software Image Management**: All workflows related to image management are now handled through Software Image Management (SWIM). SWIM offers improved management of device software images, enabling seamless deployment, upgrades, and downgrades across a two-version range. Additionally, it supports specialized firmware upgrades for Field Programmable Devices (FPD) to maintain devices with unique firmware needs efficiently..
- **Monitoring Policies**:

  Monitoring policies help you control how Crosswork monitors your network. You can create and customize different monitoring policies to monitor network-wide device information and manage your network health. Monitoring policies are available for:
  - Device Health
  - Interface Health
  - LSP Traffic Policy
  - Optical SFP Interfaces
  - Optical ZR Pluggable Devices
- **Alert Management**:

  Crosswork's alert management has been improved to offer a more comprehensive system notification experience. Enhancements include:
  - Standardized alarms and events notifications integration and visibility.
  - Option to configure and customize your settings to receive alerts.
  - System-level event processing with throttling mechanisms to prevent system overload and maintain network stability and performance.
- **Zero Touch Provisioning**:
  - **ZTP is integrated with EMF**

ZTP is now integrated with Element Management Functions (EMF), enabling deployment directly through the CAPP file.

- **UI/UX workflow changes**

  The ZTP sub-menu is integrated under the "Device Management" section of the UI main menu, consolidating various individual sub-menus.

For more information, see the *Zero Touch Provisioning* section in Cisco Crosswork Network Controller 7.0 Device Lifecycle Management.

**Documentation**

- An Information Portal is now available for Crosswork Network Controller 7.0. The information is categorized by functional area, making it easy to find and access.
- Cisco Crosswork Network Controller 7.0 Installation Guide covers installing the cluster and Crosswork applications on top of the infrastructure. It also includes installing the Cisco Crosswork Data Gateway.
- Cisco Crosswork Network Controller 7.0 Administration Guide covers the setup and maintenance of the Crosswork system. This guide also includes information on Cisco Crosswork Data Gateway and the Single VM install.
- Cisco Crosswork Network Controller 7.0 Solution Workflow Guide provides an overview of the solution and its supported use cases. It walks users through various common usage scenarios to illustrate how they can work with the solution components to achieve the desired benefits.
- The Cisco Crosswork Network Controller 7.0 Device Lifecycle Management Guide details the steps for onboarding, managing, and monitoring network devices. It covers key aspects such as alarm management, monitoring policies, ZTP and software image management (SWIM).
- Cisco Crosswork Network Controller Getting Started Guide has been deprecated, and the topics in this guide are now covered in other guides.

# Supported Use Cases

Crosswork Network Controller supports a wide range of use cases, allowing operators to manage many aspects of the network. The following use cases illustrate the most commonly used features and the applications needed to implement them. In addition, Crosswork Network Controller solution is highly adaptable and if the use case you are focused on is not covered, consult your Cisco Customer Experience representative for more information.

- **Orchestrated service provisioning:** Provisioning of layer 2 VPN (L2VPN) and layer 3 VPN (L3VPN) services with underlay transport policies to define, meet, and maintain service-level agreements (SLA) using the UI or APIs. Using Segment Routing Flexible Algorithm (Flex-Algo) provisioning and visualizing to customize and compute IGP shortest paths over a network according to specified constraints.

  For this use case, Cisco Crosswork Advantage must be installed.

- **Real-time network and bandwidth optimization:** Intent-based closed-loop optimization, congestion mitigation, and dynamic bandwidth management based on Segment Routing and RSVP-TE. Optimization of bandwidth resource utilization by setting utilization thresholds on links and calculating tactical alternate paths when thresholds are exceeded.

- **Circuit Style Segment Routing Traffic Engineering (CS SR-TE) policy provisioning with network topology visualization:**
    - Straightforward verification of CS SR-TE policy configurations
    - Visualization of CS SR-TE details, bi-directional active and candidate paths
    - Operational status details
    - Failover behavior monitoring for individual CS SR-TE policies
    - A percentage of bandwidth reservation for each link in the network
    - Manually triggered recalculations of existing CS SR-TE policy paths that may no longer be optimized due to network topology changes

  For this use case, Cisco Crosswork Advantage must be installed.

- **Local Congestion Management:** Local Congestion Mitigation (LCM) provides localized mitigation recommendations within surrounding interfaces using standard protocols. Data is gathered in real-time, and solutions are suggested when congestion is detected. LCM supports deployment as either "human in the loop" or fully automated implementations, allowing operators to choose how to use the feature. See the Local Congestion Mitigation chapter in the Crosswork Network Controller 7.0 Network Bandwidth Management guide for more information. For this use case, Cisco Crosswork Advantage must be installed.
- **Visualization of network and service topology and inventory:** The topology UI, along with the various tables that can be accessed from it, allows you to easily assess the health of the network and drill down to see details about devices, links, and services.
- **Performance-based closed-loop automation:** Automated discovery and remediation of problems in the network by allowing Key Performance Indicator (KPI) customization and execution of pre-defined remediation tasks when a KPI threshold is breached. Health Insights and Change Automation functions must be installed for this use case.
- **Planning, scheduling, and automating network maintenance tasks:** Scheduling an appropriate maintenance window for a maintenance task after evaluating the potential impact of the task (using Crosswork Planning Design). Automating the execution of maintenance tasks (such as throughput checks, software upgrades, and SMU installs) using playbooks. For this use case, Health Insights and Change Automation functions must be installed.
- **Secured zero-touch onboarding and provisioning of devices:** Onboarding new IOS-XR devices and automatically provisioning Day0 configuration resulting in faster deployment of new hardware at lower operating costs. For this use case, Cisco Crosswork Essentials must be installed.
- **Visualization of native SR paths:** Using the traceroute SR-MPLS multipath command to get the actual paths between the source and the destination can be achieved using Path Query. With the Cisco Crosswork Network Controller, a traceroute command runs on the source device for the destination TE-Router ID and assists in retrieving the paths. For this use case, Cisco Crosswork Advantage must be installed.
- **Provision, Visualize, and Analyze Tree Segment Identifier Policies in Multipath Networks:** Creating and visualizing static Tree-SID policies using the UI. Static mVPN Tree-SID policies associated with existing or newly created L3VPN service models (SR MPLS point-to-multi-point) using the Crosswork Network Controller can be visualized and analyzed to assist in efficient management and troubleshooting of your multicast network. Configuring link affinities to specify the link attributes that determine which links are suitable to form a path for the Tree-SID policy and map each bit position or attribute with a color (making it easier to refer to specific link attributes). Modifying existing static Tree-SID policies

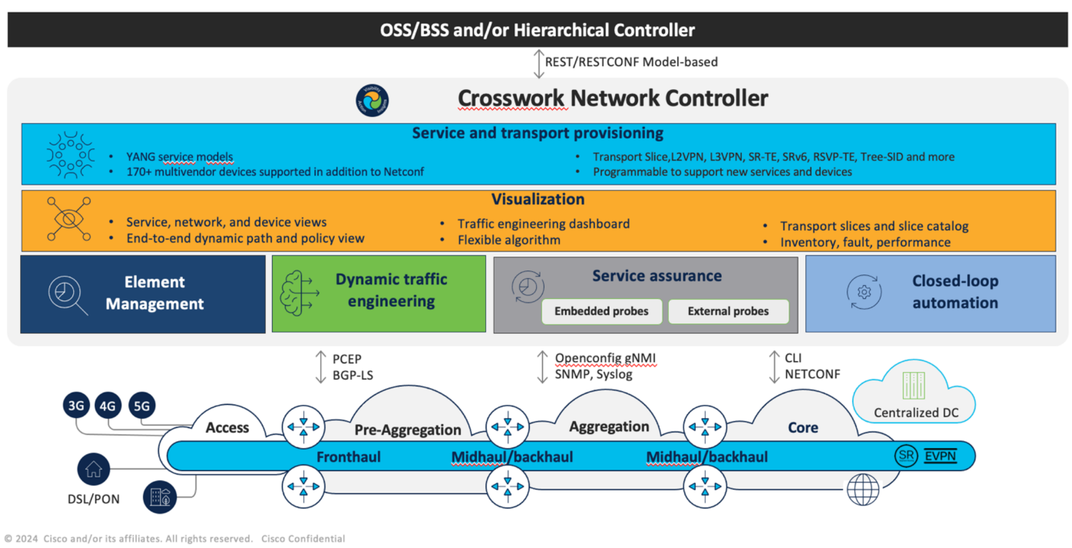
and mVPN Tree-SID policies associated with an L3VPN service model – both edit and delete – using the UI. For this use case, Cisco Crosswork Advantage must be installed.

- **Transport Slice Provisioning:** Cisco Crosswork Network Controller offers direct support for network slicing at the OSI transport layer. Using this solution, network engineering experts can design slice profiles around customer intents and add them to a catalog. Network line operators can assign the profile identified for a given customer to their endpoints and adjust the constraints according to the customer's requirements. Using the UI, you can inspect the slice details for active symptoms, failures, and root causes. In addition, the slice can be visualized on a geographical map. For this use case, Cisco Crosswork Advantage must be installed.

# Solution Components Overview and Integrated Architecture

The following diagram provides a high-level illustration of how the solution's components work together within a single pane of glass to execute the primary supported use cases.

***Figure 1: Solution Components and Integrated Architecture***




The following components make up the Cisco Crosswork Network Controller 7.0 solution:

## Cisco Crosswork Active Topology

Cisco Crosswork Active Topology's logical and geographical maps provide real-time visibility into the physical and logical network topology, service inventory, SR-TE policies, and RSVP-TE tunnels, all within a single pane of glass. They enable operators to see the status and health of the devices, services, and policies at a glance. Services and transport policies can be visualized end-to-end as an overlay within the context of the topology map. Cisco Crosswork Active Topology provides device grouping functionality so that operators can set up their maps to monitor exactly the set of devices, services, and locations for which they are responsible. In addition, operators can save custom views for quick and easy access to the views and functionality they use on an ongoing basis.

# Cisco Crosswork Optimization Engine

Cisco Crosswork Optimization Engine provides real-time network optimization, allowing operators to effectively maximize network capacity utilization, preserve network intent with proactive network monitoring and visualization, and increase service velocity. Leveraging real-time protocols, such as BGP-LS and Path Computation Element Communication Protocol (PCEP) and SR-PCE, Crosswork Optimization Engine enables near real-time tracking of the network, with the ability to react quickly (manually or through automation) to changes in network conditions to minimize disruptions or degradation in performance.

# Cisco Service Health

Service Health substantially reduces the time required to detect and troubleshoot service quality issues. It monitors the health of provisioned L2 and L3 VPN services and lets operators pinpoint why and where a service is degraded. This is accomplished through a heuristic model that provides the following:

- Monitoring the health of:
  - Point-to-point L2VPN services
  - Multipoint L2VPN (EVPN E-LAN and E-Tree L2VPN EVPN) services
  - L3VPN services
- Analysis and troubleshooting of services with degraded health
- Visualize the health status of a service and view its logical health dependency tree to help operators troubleshoot cases of degradation by locating where the problem resides, indicating possible symptoms, and impacting metrics in case of degradation
- Performance metrics and health status of Traffic Engineering (TE) policies
- Historical view and trends of service health status
- Extensible to add service monitoring capabilities to address specific needs

# Cisco Crosswork Data Gateway

Cisco Crosswork Data Gateway is a secure, standard collection platform for collecting telemetry and other performance data from compatible non-Cisco and Cisco network devices. Several data-collecting protocols, including MDT, SNMP, CLI, standards-based gNMI (dial-in), and Syslog, are supported by Crosswork Data Gateway. By doing this, it can enable a wide range of use cases and modifications. Operators can add their collection jobs to acquire network performance data, which can subsequently be sent to suitable Kafka and gRPC message buses for consumption by other applications using APIs and the configuration examples provided by Cisco. Rather than requiring each data consumer to collect information directly from the source, Crosswork Data Gateway enables the operator to capture the data once and send it to numerous consumers.

With Cisco Crosswork Network Controller operating as the controller and consumer of data and Crosswork Data Gateway working as both a centralized shared collector and distributor of data, Cisco has established a mechanism for obtaining data from the network that is reliable, flexible, and efficient.

Several Crosswork Data Gateway VMs can be installed and scaled horizontally as a pool of devices capable of handling your network's data-gathering demands to provide high availability within the pool. The number of pools and Crosswork Data Gateways in the pool is determined by the number of devices in your network, the geographic distribution of those devices, the amount of data you collect, and the level of redundancy desired (1 to 1 or n to m). For more details on scaling your Crosswork Data Gateways to match your specific use case, please collaborate with Cisco Customer Experience (CX), the Cisco account team, or the partner from whom you purchase Cisco products.

# Crosswork Common UI and API

All Cisco Crosswork Network Controller's functionality is provided within a common graphical user interface. This common UI brings together the features of all Crosswork Network Controller's components, including common inventory, network topology and service visualization, service and transport provisioning, and system administration and management functions. When optional add-on Crosswork components are installed, their functionalities are also fully integrated into the common UI. Having all functionality within a common UI instead of navigating individual application UIs separately enhances the operational experience and increases productivity.

A common API enables Crosswork Network Controller's programmability. The common APIs provide a single access point for all APIs exposed by various built-in components. The API provides a REST-based Northbound Interface to external systems (e.g., OSS systems) to integrate with Cisco Crosswork Network Controller. RESTCONF and YANG data models are made available for optimization and service provisioning use cases. For details about the APIs and examples of their usage, see the Cisco Crosswork Network Automation API Documentation on Cisco DevNet.

# Crosswork Infrastructure and Shared Services

The Cisco Crosswork Infrastructure provides a resilient and scalable platform on which all Cisco Crosswork components can be deployed. This infrastructure and shared services provide:

- A single API endpoint for accessing all APIs of Crosswork applications
- A shared Kafka bus to pass data between applications
- Shared Databases
    - Stores all configuration data for each of the applications.
    - Stores all the time series (telemetry) data gathered from the network.
- A robust Kubernetes-based orchestration layer that gives process-level resiliency and elasticity to scale the environment when additional resources are needed.
- Tools for monitoring the health of the infrastructure.

# Cisco Crosswork Health Insights and Cisco Crosswork Change Automation

Cisco Crosswork Health Insights and Cisco Crosswork Change Automation are components that can optionally be installed with Cisco Crosswork Network Controller.

Cisco Crosswork Health Insights performs real-time Key Performance Indicator (KPI) monitoring, alerting, and troubleshooting. When used with Cisco Change Automation, or as part of a broader integration with your existing automation solutions, Health Insights plays a key role in both manual and automated response to network events.

Cisco Crosswork Change Automation automates the process of deploying changes to the network. Orchestration is defined via an embedded Ansible Playbook, and then configuration changes are pushed to Cisco Network Services Orchestrator (NSO) to be deployed to the network.

These components within Cisco Crosswork Network Controller enable closed-loop discovery and remediation of network problems. Operators can match alarms to pre-defined remediation tasks, which are automatically performed when a defined Key Performance Indicator (KPI) threshold is breached. This reduces the time it takes to discover and repair a problem.

# Element Management Functions

A library of functions that provides deep inventory collection, device management, alarm management, and software image management.

Zero Touch Provisioning with automatic onboarding of new IOS-XR and IOS-XE devices and provisioning of Day0 configuration, resulting in faster deployment of new hardware at a lower operating cost.

# Cisco Network Services Orchestrator

Cisco Network Services Orchestrator (NSO) is an orchestration platform that leverages pluggable function packs to translate network-wide service intent into device-specific configuration. Cisco Network Services Orchestrator provides flexible service orchestration and lifecycle management across physical network elements and cloud-based virtual network functions (VNFs), fulfilling the role of the Network Orchestrator (NFVO) within the European Telecommunications Standards Institute (ETSI) architecture. It provides complete support for physical and virtual network elements, with a consistent operational model across both. With the ability to orchestrate across multi-vendor environments and support multiple technology stacks, Cisco Network Services Orchestrator empowers the extension of end-to-end automation to virtually any use case or device.

Cisco Network Services Orchestrator has a rich set of APIs designed to allow developers to implement service applications. It provides the infrastructure for defining and executing the YANG data models that are needed to realize customer services. It is also responsible for providing the overall lifecycle management at the network service level.

Service and device models, written using YANG modeling language, enable Cisco Network Services Orchestrator to efficiently 'map' service intent to device capabilities and automatically generate the minimum required configuration to be deployed in the network. This feature, facilitated by Cisco Cisco Network Services Orchestrator's FASTMAP algorithm, can compare current configuration states with a service's intent and then generate the minimum set of changes required to instantiate the service in the network.

All Crosswork components that are included in Cisco Crosswork Network Controller or are optional add-ons, require integration with Cisco Network Services Orchestrator.

Cisco Crosswork Network Controller requires the following Cisco Network Services Orchestrator function packs:

- SR-TE core function pack (CFP) enables the provisioning of explicit and dynamic segment routing policies, including SRv6, and on-demand SR-TE policy instantiation for prefixes with a specific color.
- The IETF-compliant L2VPN and L3VPN Core Function Packs provide baseline L2VPN and L3VPN provisioning capabilities, based on IETF NM models. Prior to customization, these sample function packs enable provisioning of the following VPN services:

**Note** The Service Health function pack should be independently installed apart from Cisco Crosswork Network Controller function packs.

- L2VPN:
    - Point-to-point VPWS using Targeted LDP
    - Point-to-point VPWS using EVPN
    - Multipoint VPLS using EVPN (with service topologies ELAN, ETREE, and Custom)
- L3VPN – both IPv4 and IPv6 address families are supported.
- Sample IETF-compliant RSVP-TE function pack intended as a reference implementation for RSVP-TE tunnel provisioning, to be customized as required.

**Note** By default, the IETF-compliant NM models are used. If your organization wishes to continue using the Flat models provided with the previous version, a manual setup process is required. Consult your Cisco Customer Experience representative for more information.

**Note** The Cisco Network Services Orchestrator sample function packs are provided as a starting point for service provisioning functionality in Cisco Crosswork Network Controller. While the samples can be used "as is" in some limited network configurations, they are intended to demonstrate the extensible design of Cisco Crosswork Network Controller. Answers to common questions can be found on Cisco Devnet, and Cisco Customer Experience representatives can answer general questions about the samples. Support for customization of the samples for your specific use cases can be arranged through your Cisco account team.

**Note** Cisco Network Services Orchestrator currently does not support bundle ethernet (BE), route distinguisher (RD), or BGP route-target (RT) functions with L2VPN EVPN. Although it does support multihoming and L2VPN route policy, there is no option to specify an RD value in L2VPN for an EVPN ELAN/ETREE, nor is there an option to specify load balancing type. To perform these functions, contact your Cisco account team for a set of custom configuration templates and advice on configuring bundles manually.

# Cisco Segment Routing Path Computation Element

Cisco Segment Routing Path Computation Element (SR-PCE) is an IOS-XR multi-domain stateful Path Computation Engine (PCE) supporting segment routing (SR), Resource Reservation Protocol (RSVP), and

SRv6-aware PCE. Cisco Segment Routing Path Computation Element builds on the native PCE abilities within IOS-XR devices and provides the ability to collect topology and segment routing IDs through IGP (OSPF or IS-IS) or BGP Link-State (BGP-LS), calculate paths that adhere to service SLAs, and program them into the source router as an ordered list of segments. A Path Computation Client (PCC) reports and delegates control of head-end tunnels sourced from the PCC to a PCE peer. The PCC and PCE establish a Path Computation Element Communication Protocol (PCEP) connection that Segment Routing Path Computation Element uses to push updates to the network and re-optimize paths where necessary. PCEPv6 is also supported.

Cisco Segment Routing Path Computation Element can either reside on server resources using virtualized XRv9000 , or run as a converged application within IOS-XR Routers.

# Multi-Vendor Capabilities

Crosswork Network Controller is multivendor capable, leveraging open industry standard mechanisms and protocols such as BGP-LS, SNMP, gNMI, PCEP, segment routing, and NETCONF/YANG to communicate with network devices in a multivendor environment. In order to deploy the product in a multivendor environment, Cisco professional services (CX) should be engaged to validate interoperability with third-party devices in your network environment. See the Cisco Crosswork Network Controller Data Sheet for supported use-cases and capabilities.

Today’s networks have typically been built over time and incorporate multiple vendors and generations of hardware and software. Furthermore, there is a lack of industry standardization, making support for these networks using a single tool challenging.

Service providers require an integrated solution to manage third-party devices that will reduce operational expenses and maintenance overhead, as well as eliminate the need to build custom applications to deploy and maintain different vendor products for a single network.

Using standards-based protocols, the Cisco Crosswork Network Controller has multi-vendor capabilities for:

- Network service orchestration via Cisco Network Services Orchestrator using CLI and Netconf/YANG. Cisco Network Services Orchestrator is a YANG model-driven platform for automating provisioning, monitoring, and managing applications and services across multi-vendor networks.
- The Cisco Crosswork Network Controller provisioning functionality can be extended using the application programming interfaces (APIs). Each product in the platform supports external integration, development, and customization by providing easy-to-use APIs that cover all or most of each product's functions, including functions created exclusively for access via APIs. For more information, see the Cisco Crosswork Network Automation API Documentation on Cisco DevNet.
- Telemetry data collection via the Cisco Crosswork Data Gateway using SNMP with standards-based MIBs, Syslog, gNMI, and CLI commands. Cisco Crosswork Data Gateway also supports Native YANG data models for external destinations and SNMP MIBs. Custom packages are available to use with Crosswork applications, such as Crosswork Health Insights, for device telemetry and network management automation.
- Topology and transport discovery via SR-PCE, using IGP and BGP-LS, with link utilization and throughput collected via SNMP using standard MIBs.
- Transport path computation using PCEP.

Building a custom package (or modifying the samples we provide) can get complicated. Refer to the Cisco DevNet guide to get details about the process. This documentation includes the steps to load custom packages and the basic steps needed to leverage them. Even with these extensive resources, operators may find it more productive to use the expertise from Cisco CX (Cisco Customer Experience) to perform this work. For more details, contact Cisco or the Cisco partner you work with to purchase products and services.

CHAPTER 2

# UI Overview

This section explains the following topics:



# Log In

Log into the web UI by entering the following URL in the browser's address bar:

```
https://<Crosswork Management Network Virtual IP (IPv4)>:30603/
```

```
https://[<Crosswork Management Network Virtual IP (IPv6)>]:30603/
```

**Note** The IPv6 address in the URL must be enclosed with brackets.

In the Log In window, enter the username and password configured during installation and click **Log In**.

Self-signed certificate: At first-time access, some browsers display a warning that the site is untrusted. When this happens, follow the prompts to add a security exception and download the self-signed certificate from the server. After downloading the certificate, the browser accepts the server as a trusted site in all future login attempts.

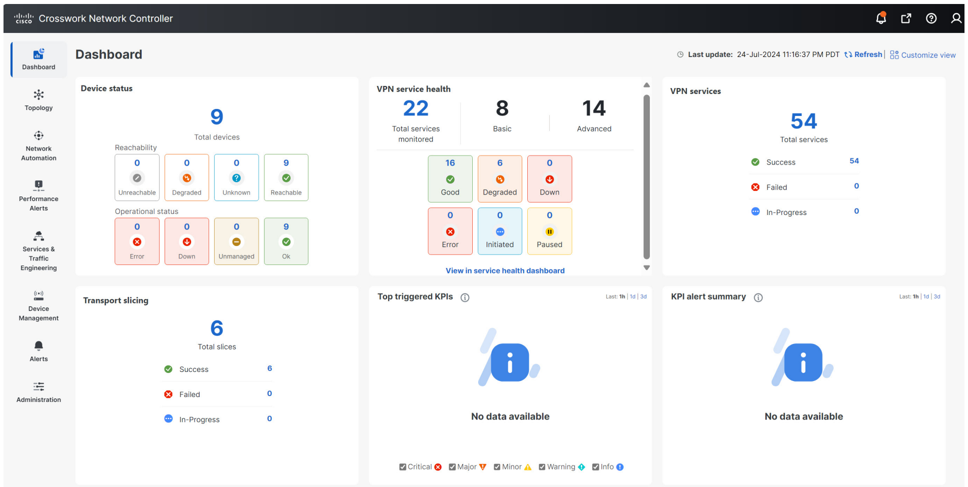
CA signed certificate: For production use, a CA signed certificate may be installed and is recommended to avoid a warning that the site is untrusted.

**Note** For information on installing CA signed certificates, see the Manage Certificates topic in *Crosswork Network Controller Administration Guide*.

# Dashboard

After successful login, the Home page opens. The Home page displays the dashboard, providing an at-a-glance operational summary of the managed network. The dashboard is made up of a series of dashlets. The specific dashlets included in your dashboard depend on which Cisco Crosswork applications you have installed. Links in each dashlet allow you to drill down for more details.

***Figure 2: Dashboard Home Page***




> **Note** Your Dashboard may differ from this screen capture, which displays optional components you may not have installed.

# Navigation

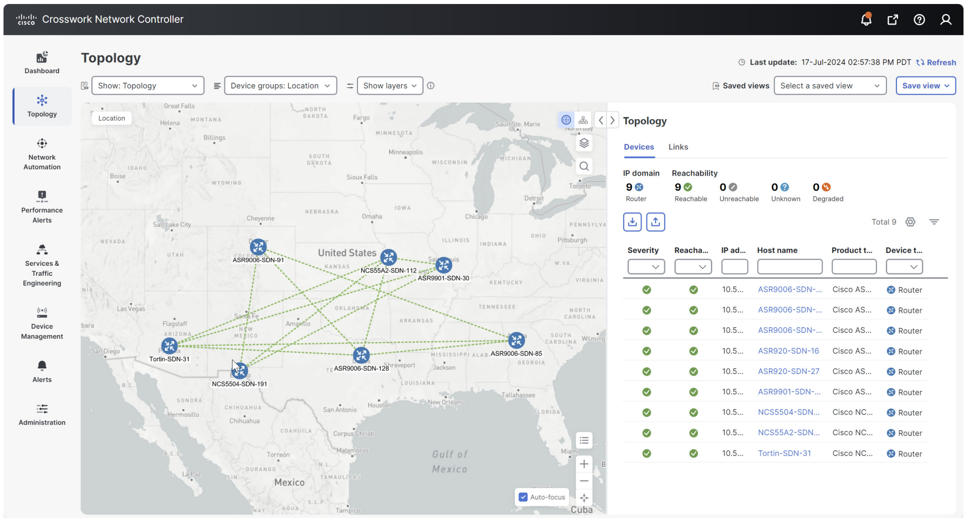
The main menu along the left side of the window provides access to all features and functionality in the Crosswork Network Controller, as well as to device management and administrative tasks. The Dashboard, Topology, Services & Traffic Engineering, Device Management, and Administration menu options are available when all native Crosswork Network Controller components are installed. Additional menu options are available in the main menu depending on which Cisco Crosswork add-on applications are installed.

**Dashboard**

The home page contains the dashboard, as described in the Dashboard topic.

**Topology**

*Figure 3: Topology Page*

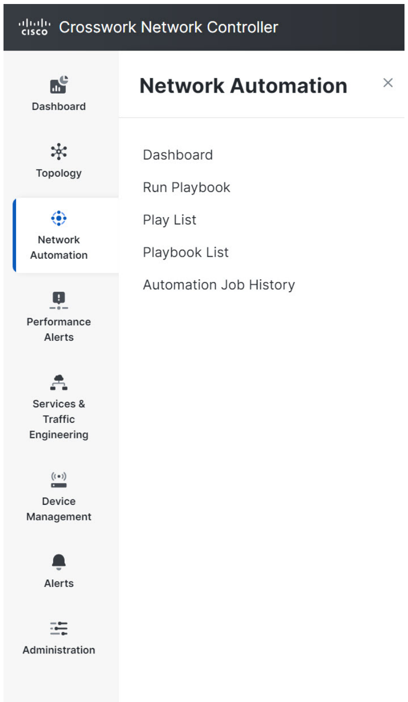



Users can display the network device and link topology on a geographical (geo) or logical map. The geo map shows single devices, device groups, clusters, links, and tunnels superimposed on a world map. Each device's location on the map reflects its GPS coordinates (longitude and latitude). Operators supply this location information in CSV or KML files uploaded using Cisco Crosswork UI or Device UI. The logical map shows devices and their links positioned according to a user-selected algorithm without considering their physical location.

The Topology page offers options to customize the data display based on the installed applications. You can select Topology, Traffic Engineering, VPN Services, or Transport Slicing using the Show drop-down list. The map can be filtered to display device groups, devices, and links for specific layers. In addition to quickly visualizing device status and health, you can view detailed information about a device and its associated links by selecting the device in the table or on the map. A global topology search can also be done using device names, location, or civic location.

### Network Automation

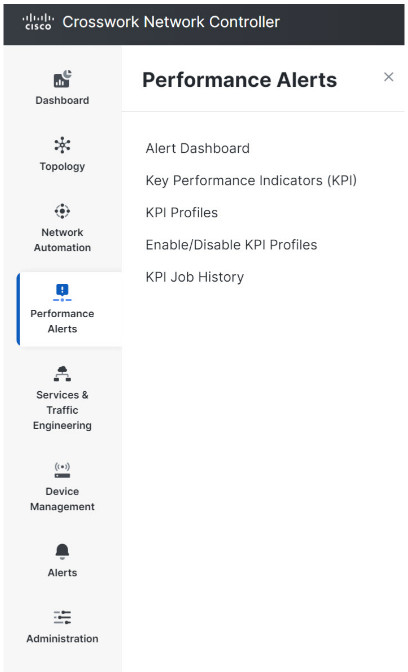
*Figure 4: Network Automation Menu*




The Network Automation menu facilitates the automation of deploying network changes. Using Playbooks, which are comprised of YAML-written Plays, you can define automation tasks to achieve the desired network states. The Dashboard provides an overview of all Playbook-related activity, including pre-defined and custom Playbooks, and allows for the initiation of Playbook runs. Playbooks can be executed in various modes, such as Dry Run, Single Stepping, or Continuous, to align with specific requirements. Furthermore, you can view, create, import, export, and manage custom Plays and Playbooks as needed. It is also possible to schedule future Playbook runs as jobs and monitor their progress for diagnosing failures. Additionally, Crosswork Network Controller allows for the manual execution of KPI-linked Playbooks and the automatic running of one or more KPI-linked Playbooks when the associated KPI raises an alert of sufficient severity.

### Performance Alerts

*Figure 5: Performance Alerts Menu*

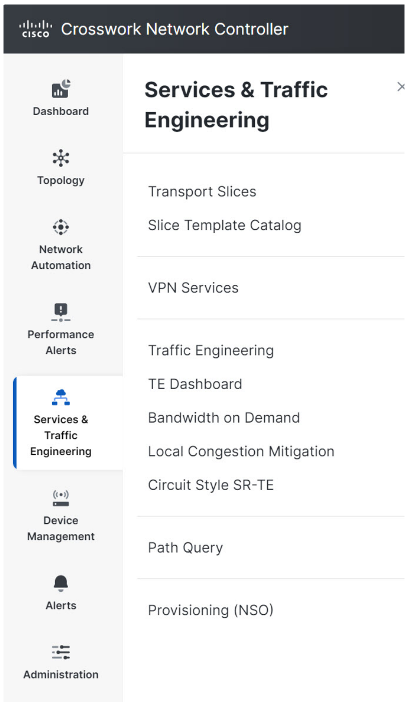



The Performance Alerts menu offers real-time, telemetry-based Key Performance Indicator (KPI) monitoring and intelligent alerting. The alerts are based on predefined templates or user-defined logic and can be tied to the Playbooks to implement closed-loop automation workflows. The Alert Dashboard offers a summary of device health information based on real-time network state events. With Performance Alerts, you have a complete access to Cisco-supplied and user-created KPIs. You can also create and manage KPI Profiles by grouping KPIs and configuring parameters relevant to monitoring specific types of devices based on their purpose (for example, environmental or health check). You can also enable or disable KPI profiles on desired devices and monitor the progress of each KPI job deployment.

You can also link your KPIs to the Change Automation application's Playbooks.

**Services & Traffic Engineering**

*Figure 6: Services & Traffic Engineering Menu*



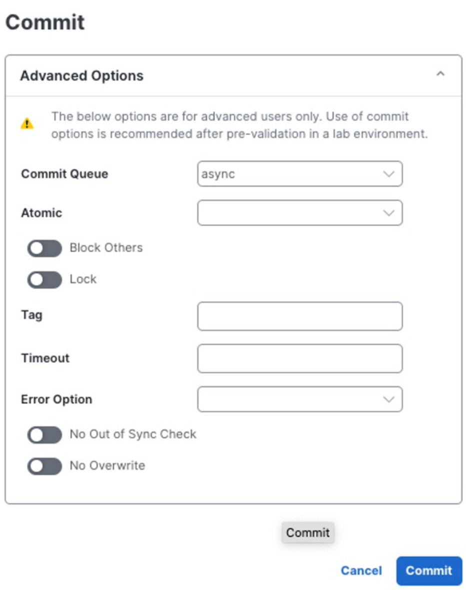

The Services & Traffic Engineering menu provides access to VPN and transport provisioning and visualization functionality, bandwidth management functionality, and access to the configuration pages used to enable Feature Packs. For more information, see Crosswork Network Controller 7.0 Traffic Engineering and Optimization Guide.

Choose **VPN services** or **Traffic Engineering** to see managed VPN services, SRv6 policies, or SR-TE policies/RSVP-TE tunnels within the context of a logical or geographical map.

Choose **Provisioning (NSO)** to access the provisioning UI rendered from the Cisco Network Services Orchestrator models. Here, you can create L2VPN and L3VPN services, SR-TE policies, SRv6 policies, SR ODN templates, and RSVP-TE tunnels. You can also create the resources required for these services and policies, such as resource pools, route policies for L2VPN and L3VPN services, and SID lists for SR-TE policies. SR-TE policies and RSVP-TE tunnels can be attached to VPN services to define and maintain SLAs by tracking network changes and automatically reacting to optimize the network.

Before committing a configuration, **Commit – Advanced Options** may be available. These options are designed for advanced users only. Use of the commit options is recommended after pre-validation in a lab environment. In the Advanced Options window, the following options are available:

*Figure 7: Advanced Options Window*

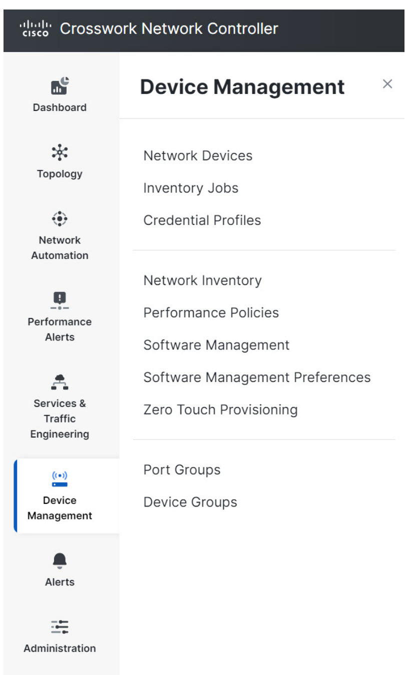



- **Commit Queue**: While the configuration change is committed to the configuration database (CDB) immediately, it is not committed to the actual device but queued for eventual commit to increase transaction throughput. This enables the use of the commit queue feature for individual commit commands without enabling it by default. There are two operation modes: async and sync.
  - **async mode**: The async mode operation returns successfully if the transaction data has been successfully placed in the queue.
  - **sync mode**: The sync mode will prevent the operation from returning until the transaction data has been sent to all devices or a timeout occurs. If the timeout occurs, the transaction data stays in the queue, and the operation returns successfully. The timeout value can be specified with the timeout or infinity option. By default, the timeout value is determined by what is configured in /devices/global-settings/commit-queue/sync.
- **Atomic**: The atomic option sets the atomic behavior of the resulting queue item. If this is set to false, the devices contained in the resulting queue item can start executing if the same devices in other non-atomic queue items ahead of it in the queue are completed. If set to true, the atomic integrity of the queue item is preserved.
  - **Block Others**: This option will cause the resulting queue item to block subsequent queue items that use any of the devices in this queue item from being queued.
  - **Lock**: This option will lock the resulting queue item. The queue item will not be processed until it has been unlocked; see the actions unlock and lock in /devices/commit-queue/queue-item. As long as the lock is in place, no following queue items using the same devices will be allowed to execute.
- **Tag**: This option sets a user-defined, opaque tag that is present in all notifications and events sent referencing the queue item..
- **Timeout**: The timeout value can be specified with the timeout or infinity option. By default, the timeout value is determined by what is configured in /devices/global-settings/commit-queue/sync.

- **Error Option**: Depending on the Error Option selected, NSO will store the reverse of the original transaction to undo the transaction changes and return to the previous state. This data is stored in the /devices/commit-queue/completed tree, which it can be viewed and invoked with the rollback action. When invoked, the data will be removed. There are two values available: continue-on-error, stop-onerror.
    - **continue-on-error:** The continue-on-error value means that the commit queue will continue on errors. No rollback data will be created.
    - **stop-on-error**: The stop-on-error means that the commit queue will lock the failed queue item, thus blocking other queue items with overlapping devices from being executed. The lock must be released manually when the error is fixed, or the rollback action must be invoked under /devices/commit-queue/completed.
- **No Out of Sync Check**: Commit even if out of sync.
- **No Overwrite**: Do not overwrite modified data on the device.

### Device Management

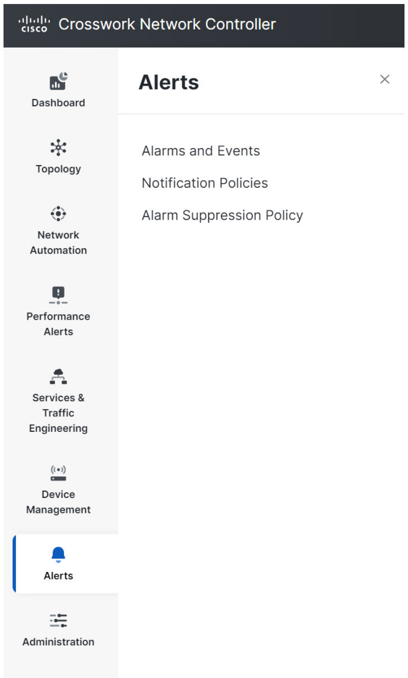
***Figure 8: Device Management Menu***




The Device Management menu provides access to device-related functionality, including adding, managing, and grouping devices, creating and managing credential profiles, and viewing a history of device-related jobs.

### Alerts

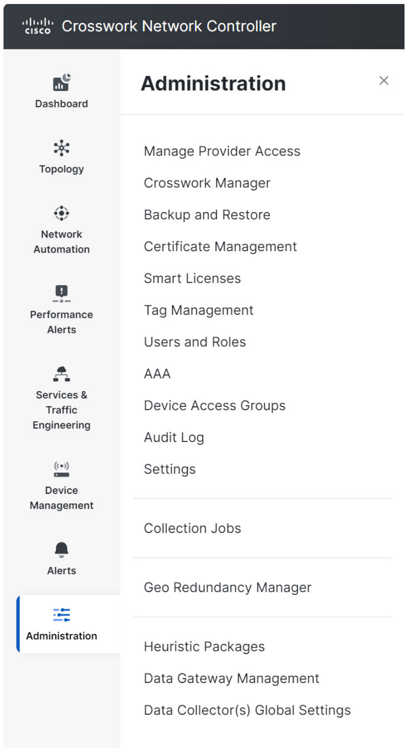
*Figure 9: Alerts Menu*




The Alerts menu gives you a quick summary of all the alarms and events in your system and network. By configuring your alarms and events, you can effectively manage system performance and promptly address issues. The Notification Policies enable you to create new alarm notification policies that will send the alarms generated by Crosswork Network Controller to a Northbound trap or a Syslog receiver based on specific criteria. If you need to temporarily stop notifications for specific conditions, such as during system maintenance or for known issues, you can put an alarm suppression policy in place to prevent unnecessary alerts.

### Administration

*Figure 10: Administration Menu*




The Administration menu provides access to all system management functions, data gateway management, Crosswork cluster and application health, backup and restore, smart licensing, and other setup and maintenance functions that an administrator typically performs.

See Crosswork Network Controller 7.0 Administration Guide for information about these functions.

CHAPTER 3

# Orchestrated Service Provisioning

This section explains the following topics:



## Overview

Using the scenario workflows described in this section, we provide examples of how to configure the system to deliver the operator's intended configuration. These scenarios only partially demonstrate all of Crosswork Network Controller's capabilities. They are intended to demonstrate the flexibility of the platform. Additional customization is possible by leveraging the resources available on Cisco DevNet or through engagement with Cisco Customer Experience.

**Note** Some scenario features and functions belonging to multiple components (such as Crosswork Optimization Engine, Crosswork Service Health, Crosswork Active Topology) will not be available as described unless all of the applications are successfully deployed.

To illustrate the capabilities of Crosswork Network Controller, we have provided a set of scenarios we walk through to guide you in configuring the system.

**Scenarios**

Scenario: Implement and maintain SLA for an L3VPN service for SR-MPLS (using ODN)

In this scenario, we provision an L3VPN service using Segment Routing (SR) On-Demand Next Hop (ODN) to achieve specific SLA objectives such as lowest latency and disjoint paths. Additionally, we will enable Service Health to analyze and monitor the provisioned services.

Scenario: Implement and maintain SLA for an L3VPN service for SRv6 (using ODN)

This scenario focuses on provisioning the L3VPN service using SR-TE ODN with SRv6 enabled.

Scenario: Mandate a static path for an EVPN-VPWS service using an explicit MPLS SR-TE policy

This scenario demonstrates the use of pre-defined templates to create a static path for mission-critical traffic. We will show how to quickly and easily create SR-TE policies and VPN services by uploading an XML containing all the necessary configurations. Additionally, we will use Service Health to analyze and monitor the provisioned services.

Scenario: Provision an L2VPN service over an RSVP-TE tunnel with reserved bandwidth

This scenario focuses on bandwidth reservation to facilitate continuous stream transmission. We will demonstrate how to reserve bandwidth using RSVP-TE Tunnels and attach them to an L2VPN service to meet high-quality of service requirements for continuous streaming of rich media data.

Scenario: Provision a soft bandwidth guarantee with optimization constraints

This scenario builds on the previous one, with the addition of BWoD configuration to reserve bandwidth on specific links to ensure a dedicated path.

**Additional Resources**

- For information about segment routing and segment routing policies, see Crosswork Network Controller 7.0 Traffic Engineering and Optimization Guide.
- Cisco Network Services Orchestrator documentation is included in the latest Network Services Orchestrator image.

# Scenario: Implement and maintain SLA for an L3VPN service for SR-MPLS (using ODN)

This scenario walks you through the procedure for provisioning an L3VPN service with a specific SLA objective: all traffic for this service must take the lowest-latency path. The customer requires this low-latency path for this service, as all of this service's traffic is a high priority. The customer also wants to use disjoint paths, that is, two unique paths that steer traffic from the same source to two unique destinations, avoiding common links so there is no single point of failure.

We'll achieve this using Segment Routing (SR) On-Demand Next Hop (ODN). SR ODN allows a service headend router to automatically instantiate an SR-TE policy to a BGP next-hop when required (on-demand). We configure the headend with an ODN template with a specific color that identifies the SLA. Crosswork will optimize the traffic path when it receives a prefix with that SLA-specific color. We define prefixes in a route policy that is associated with the L3VPN.

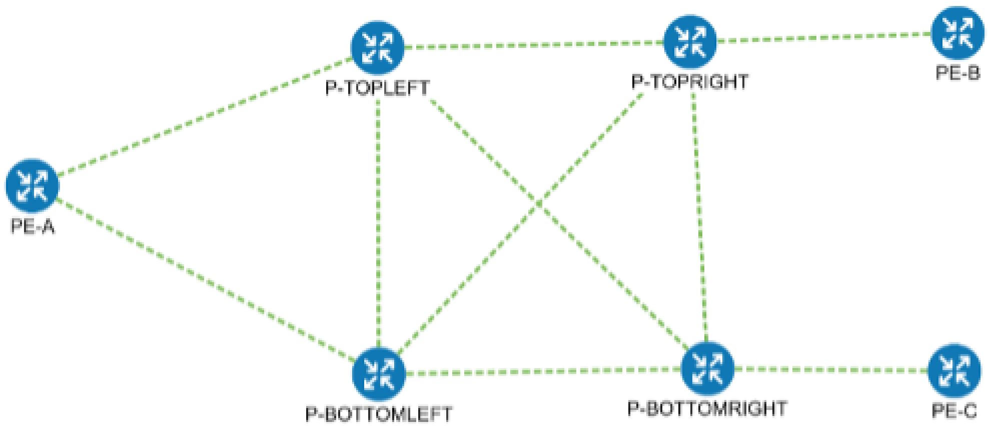
Crosswork Network Controller continues to monitor the network and, in a closed loop, automatically optimizes it based on the defined SLA.

Within this workflow, we also have the option to enable Crosswork's Service Health monitoring and to use Flex-Algo as a constraint on how paths are computed and visualized. With Service Health monitoring, operators can gather quick insights into degraded and down services and then use these insights to visualize, inspect, and troubleshoot for improved network optimization.

With Flex-Algo, we can customize IGP shortest-path computations using the algorithms we define. IGP will compute paths based on a user-defined combination of metric types and constraints and present a filtered topology view based on our specific Flex-Algo definitions.

The following topology provides the base for this scenario:

*Figure 11: Example topology for the scenario*




In this scenario, we will:

- Create a segment routing ODN template with a specific color on the endpoints to ensure that traffic is transported within an LSP (underlay) and that a best-path tunnel is created dynamically when a prefix with the specified color is received. The ODN template defines the SLA on which you want to optimize the path. In this case, we will optimize on latency.
- Specify that the computed paths are disjoint: they will not share the same link.
- Create a route policy on each endpoint to be used to bind the L3VPN to the ODN template. This route policy adds a color attribute to the customer prefixes and advertises via BGP to other endpoints. This color attribute indicates the SLA required for these prefixes.
- Create an L3VPN service with three endpoints and enable Service Health monitoring.
- Visualize how this overlay/underlay configuration optimizes the traffic path and automatically maintains the SLA while monitoring your service's health.

### Assumptions and prerequisites

- To use ODN, BGP peering for the prefixes must be configured between the endpoints or PEs. Usually, for L3VPN, this is the VPNv4 and VPNv6 address family peering.
- For Service Health enablement, Service Health must be installed. See Crosswork Network Controller 7.0 Installation Guide chapter, Install Crosswork Applications.
- Before using the Assurance Graph in Service Health, ensure that the topology map nodes are fully configured and created with a profile associated with the service. Otherwise, the Subservice Details metrics will indicate that no value has been reported yet. See Crosswork Network Controller 7.0 Service Health Guide for further details.
- L3VPN service monitoring supports XR devices but does not support XE devices. As a result, after creating an L3VPN service and enabling Service Health monitoring, the service monitoring may remain in a degraded state with a METRIC_SCHEDULER error if a provider and devices are removed and added back. To resolve this, service monitoring must be stopped and restarted.
- (Optional) Flexible Algorithms and the IDs used must be configured in your network.

**Note** Screen captures showing services and data are for example purposes only and may not always reflect the devices or data described in the workflow content.

# Step 1 Create an ODN template to map color to an SLA objective and constraints

Disjointness constraints work by associating a disjoint group ID with the ODN template. Tunnels with the same disjoint group ID will be disjoint, i.e., they will use different links, nodes, and shared risk link groups depending on how the disjoint groups are configured.

We will create the following ODN templates:

- Headend PE-A, color 72, latency, disjoint path (link), group ID 16 - L3VPN_NM-SRTE-ODN_72-a
- Headend PE-A, color 71, latency, disjoint path (link), group ID 16 - L3VPN_NM-SRTE-ODN_71-a
- Headend PE-B and PE-C, color 70, latency - L3VPN_NM-SRTE-ODN_70
- Headend PE-B, color 72, latency - L3VPN_NM-SRTE-ODN_72-b
- Headend PE-C, color 71, latency - L3VPN_NM-SRTE-ODN_71-c

For example purposes, we will show how to create the first ODN template - L3VPN_NM-SRTE-ODN_72-a. The other ODN templates can be created using the same procedure.

### Before you begin

In this step, we will create an ODN template on each endpoint. The ODN template specifies the color and the intent; in this case, latency and disjointness. This ODN template will be used to dynamically create tunnels (on-demand) when prefixes with matching colors are received via BGP. Traffic to these prefixes will be automatically steered into the newly created tunnels, meeting the SLA objective and constraints intended for these prefixes and signaled using colors in the BGP routes.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **SR-TE** > **ODN-Template**.

**Step 2** Click to create a new template and give it a unique name. In this case, the name is **L3VPN_NM-SRTE-ODN_72-a**. Click **Continue**.

You may also browse for an existing template on your system to import the file. The information from the imported file is populated into the form.

*Figure 12: SR-TE ODN-Template Page*


**Step 3** Choose the head-end device, **PE-A**, and specify the color **72**.

**Step 4** Under dynamic, select **latency** as the metric type. This is the SLA objective on which we are optimizing.

**Step 5** Select the **pce** check box to specify that the path should be computed by the SR-PCE, not by the Path Computation Client (PCC).

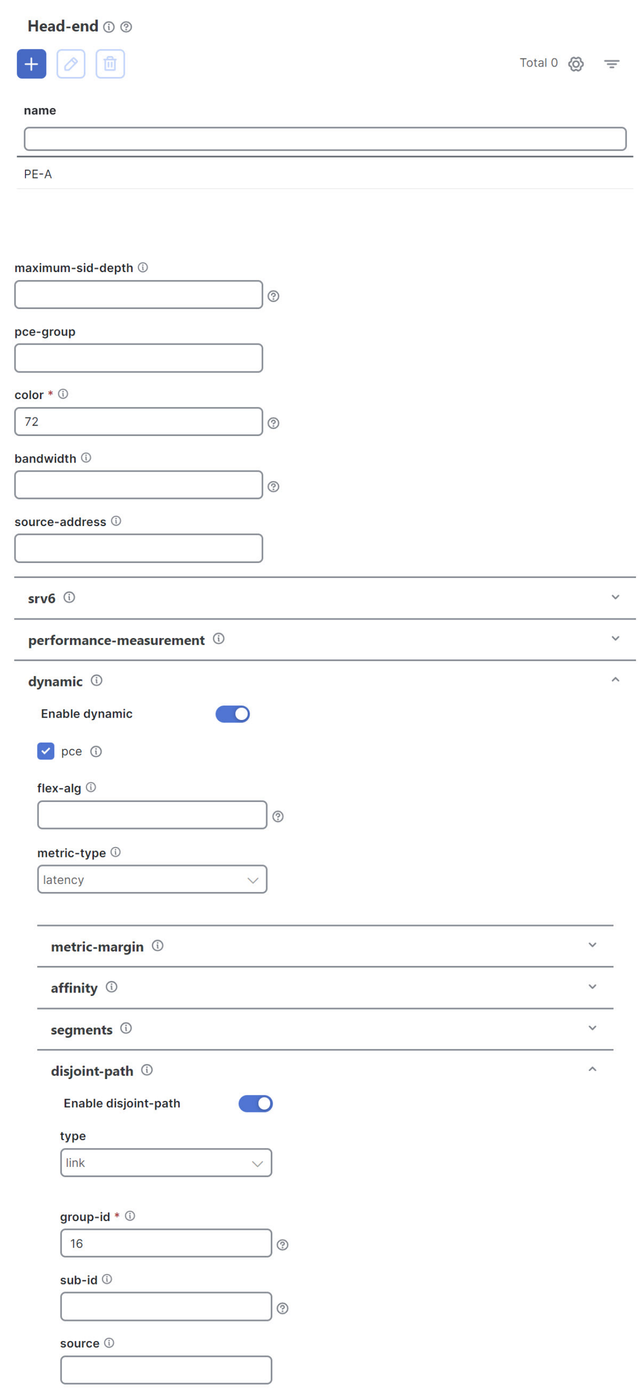
**Step 6** Define the required constraints. In this case, we want the computed paths to be disjointed because they must not share a link.

Under disjoint-path, choose **link** as the type, and specify a numeric group ID, in this case, **16**, as the group-id.

**Note** You may choose the group ID. All paths requested with the same group-id will be disjointed.

**Note** Optionally, you may configure Flex-Algo as a constraint.

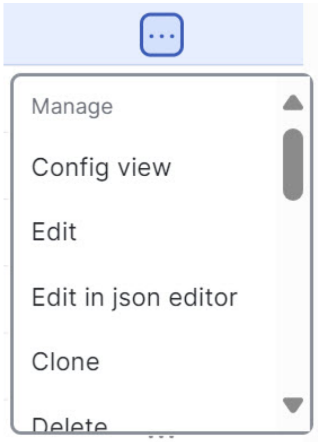
*Figure 13: SR-TE ODN-Template*




**Step 7** Commit your changes or click **Dry Run** to check what will be configured on the devices before you commit.

**Step 8** Check that the new ODN template appears in the table and its provisioning state is **Success**. Click ⋯ in the Actions column and choose **Config View** to see the Yang model-based service intent data that details the ODN template you created.

*Figure 14: Actions Column Menu*




**Step 9** Create the other ODN templates listed above in the same manner.

**Note** You can save some time by using the Clone function to build the other policies needed to complete this scenario. Simply select **Clone** from the ⋯ Actions column, provide a new name for the clone, edit the values, and then select **Commit**.

# Step 2 Create an L3VPN route policy

In this step, we will create a route policy for each endpoint, and we will specify the same color as defined in the ODN template. The route policy defines the prefixes to which the SLA applies. When traffic from the specified network with a matching color is received, paths are computed based on the SLA defined in the ODN template. We will create the following route policies by first setting the routing policy tag and routing policy destination prefix. The routing policy prefixes should match with the subnet prefix configured on the PE devices in the service:

- Color 70, IPv4 prefix 70.70.70.0/30 - L3VPN_NM-SRTE-RP-PE-A-7
- Color 71, IPv4 prefix 70.70.71.0/30 - L3VPN_NM-SRTE-RP-PE-B-7
- Color 72, IPv4 prefix 70.70.72.0/30 - L3VPN_NM-SRTE-RP-PE-C-7

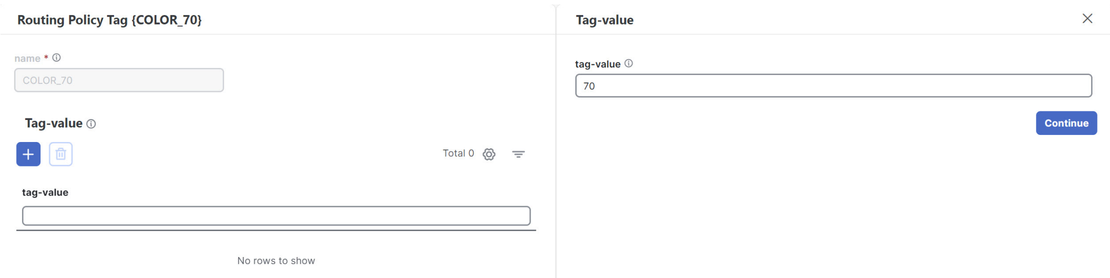
**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **Routing Policy Tag**.

**Step 2** Click + to create a new routing policy tag and type the name of the tag set: **COLOR_70**. Click **Continue**.

This is used as a label to reference the set in actions and conditions.

**Step 3** Under Tag-value, click + and type the tag-value: **70**.

*Figure 15: Routing Policy Tag*




The tag value may be a number between **1 – 4294967295** and should match to a color value.

**Step 4** Click **Continue**. The new routing policy tag name with the new tag value is visible. Click **Commit changes**.

Create the other two routing policy tags (**COLOR_71** and **COLOR_72**) and tag values (**71** and **72**) by following the same steps above. Click **Continue**.

**Step 5** Now, create the routing policy destination prefixes. From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **Routing Policy Destination Prefix.**

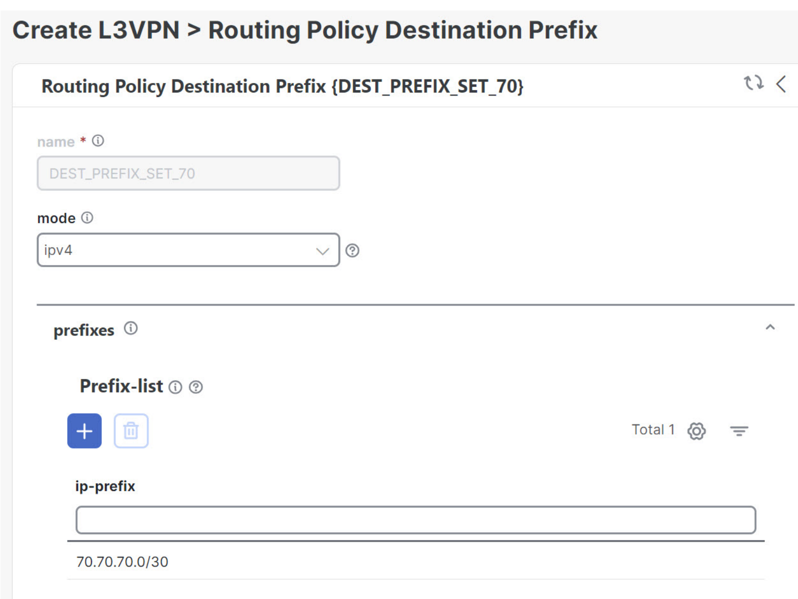
**Step 6** Click + to create a new routing policy destination prefix and type the name **DEST_PREFIX_SET_70**.

The name of the prefix set will reference the set in match conditions.

**Step 7** For Mode, select **ipv4**.

**Step 8** Expand prefixes and click + to add the ip-prefix to the prefix-list. Type **70.70.70.0/30** and click **Continue**.

*Figure 16: Routing Policy Destination Prefix*




**Step 9** Create the other two routing policy destination prefixes (**DEST_PREFIX_SET_71** and **DEST_PREFIX_SET_72**) by following the same steps.

**Step 10** Now, we are ready to create the first route policy - L3VPN_NM-SRTE-RP-PE-A-7. The other route policies can be created using the same procedure. From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3vpn** > **Routing Policy**.

**Step 11** Click + to create a new route policy and type a unique name for the top-level policy definition: **L3VPN_NM-SRTE-RP-PE-A-7**. Click **Continue**. The statements section appears.

**Note** The Route Policy statement defines the condition and actions taken by the system.

**Step 12** Expand statements, click + to add the name of the policy statement (such as **stmt1**) and click **Continue**. The statement{stmt1} panel appears, showing **conditions** and **actions** sections.

**Step 13** Expand conditions and then expand match-dest-prefix-set. In the prefix-set list, select or type the following: **DEST_PREFIX_SET_70**. This is what references a defined prefix set.

**Note** Once selected, the **Enable match-dest-prefix-set** toggle, which will match a referenced prefix-set according to the logic defined in the match-set-options list, switches on.

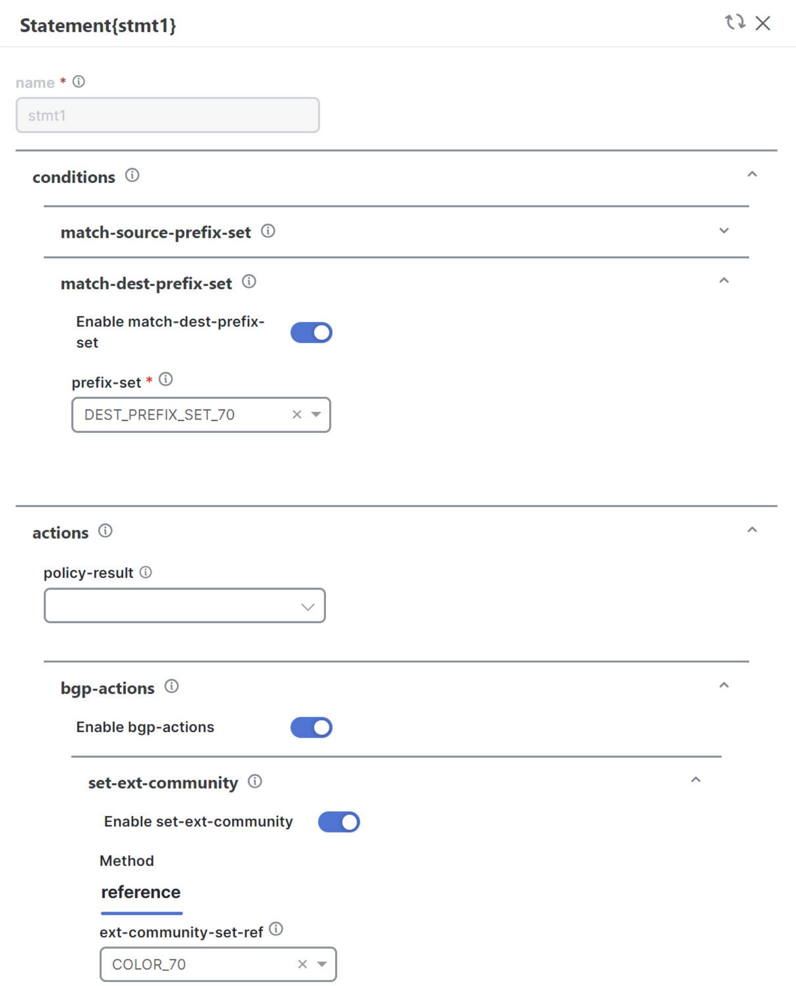
**Step 14** Expand actions and then expand bgp-actions.

**Step 15** For bgp-actions, slide the Enable bgp-actions toggle to the on position. Toggling on bgp-actions defines the top-level container for BGP-specific actions.

**Step 16** Now expand set-ext-community. Slide the Enable-set-ext-community toggle to the on position. Toggling on set-ext-community sets the extended community attributes.

**Step 17** For Method and reference, select the Ext-community-set-ref list and select **COLOR_70**. The Ext-community-set-ref references a defined extended community set by name.

*Figure 17: Route Policy Statement*




**Step 18** Click **X** in the top-right corner to close the statement{stmnt1} panel and click **Commit changes**.

**Step 19** Create the other route policies (**L3VPN_NM-SRTE-RP-PE-B-7** and **L3VPN_NM-SRTE-RP-PE-C-7**) in the same manner.

---

After creating the L3VPN route policies, create the VPN profile for each route policy and then create and provision the L3VPN service. The VPN profile will be referenced from the L3VPN service. This will bind the route policy to the L3VPN service.

# Step 3 Create and provision the L3VPN service

In this step, we will create the L3VPN service with three endpoints: PE-A, PE-B, and PE-C. Each endpoint will be associated with a vpn-instance-profile, which in turn points to a VPN profile that contains the route policy with the same color as specified in the ODN template. In this way, traffic that matches the specified prefixes and color will be treated according to the SLA specifications.

First, we will create the VPN profiles. The newly created VPN profiles will have the same names as the L3VPN routing policy names.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **VPN Profiles**.

**Step 2** Click + to create a VPN profile to be referenced in the VPN service.

**Step 3** Select the Id list and select **L3VPN_NM-SRTE-RP-PE-A-7**.

Now, create and provision the L3VPN service.

**Step 4** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3vpn** > **L3vpn-Service**.

**Step 5** Click + to create a new service and type a new vpn-id: **L3VPN_NM-SRTE-ODN-70**. Click **Continue**.

A VPN identifier uniquely identifies a VPN and has a local meaning (for example, within a service provider network).

**Step 6** Create vpn-instance-profile, which is a container that defines the route distinguisher (RD), route targets, and the export/import route policy. We will create vpn-instance-profiles for each endpoint, as follows:

- L3VPN_NM_SR_ODN-IE-PE-A-7 with route distinguisher 0:70:70
- L3VPN_NM_SR_ODN-IE-PE-B-7 with route distinguisher 0:70:71
- L3VPN_NM_SR_ODN-IE-PE-C-7 with route distinguisher 0:70:72

**a.** Expand vpn-instance-profiles and click + to create a new vpn-instance-profile profile-id: **L3VPN_NM_SR_ODN-I-PE-A-7**. Click **Continue**.

**b.** Enter the route distinguisher (rd) to differentiate the IP prefixes and make them unique. For this scenario, we are using **0:70:70**.

**c.** For address-family, click + and select **ipv4** from the list. Click **Continue**.

**d.** Define the required VPN targets, including id, route-targets, and route-target-type (import/export/both).

**e.** Under vpn-policies, in the export-policy list, choose the relevant VPN profile (which contains the route policy: **L3VPN_NM-SRTE-RP-PE-A-7**). This forms the association between the VPN and the ODN template that defines the SLA.

**f.** Click **X** in the top-right corner when you are done.

**g.** Similarly, create the other vpn-instance-profiles.

**Step 7** Define each VPN endpoint: PE-A, PE-B, and PE-C.

a) Expand vpn-nodes and click + to select the relevant device from the list: **PE-A**. Click **Continue**.

b) Enter the local-as number for network identification: **200**.

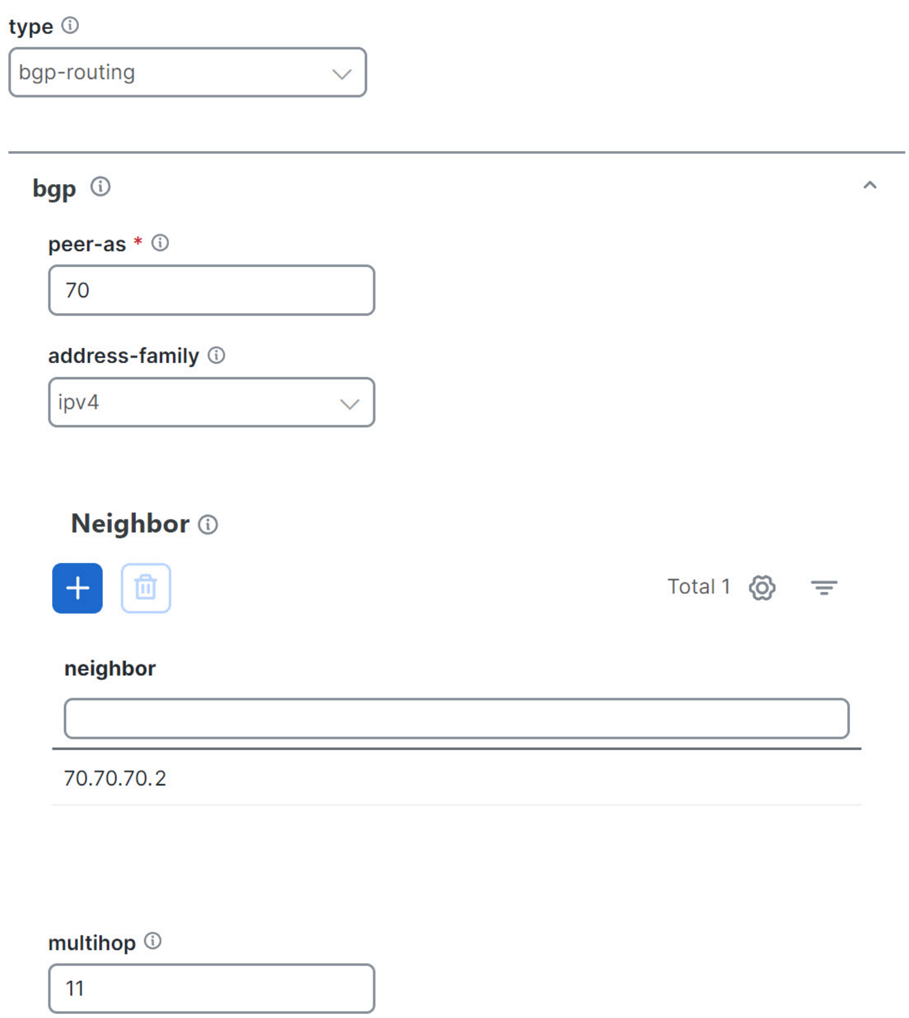
c) Expand active-vpn-instance-profiles and click + to select the profile-id you created in the previously: **L3VPN_NM-SRTE-RP-PE-A-7**. Click **Continue**.

d) Define the network access parameters for communication from the PE to the CE:

- Under vpn-network-accesses, click + to create a new set of VPN access parameters and provide a unique ID. Click **Continue**.

- In the Interface-id field, type **Loopback70**. This is the identifier for the physical or logical interface. The identification of the sub-interface is provided at the connection level and/or the IP connection level.
- Expand ip-connection > ipv4 and enter a local-address (**70.70.70.1**) and then prefix-length (**30**).
- For routing protocols, click + to create a unique id, set the type to bgp-routing, and then expand bgp to set the peer-as number (**70**), and the address-family (**ipv4**). In addition, set the bgp neighbor (**70.70.70.2**) and the multihop number (for example, **11**) that indicates the number of hops allowed between the bgp neighbor and the PE device.

***Figure 18: L3vpn-Service Routing-protocol***




- Click **X** in the top-right corner until you are back on the Create L3VPN screen.
- Similarly, create the other VPN nodes: **PE-B** and **PE-C**.

**Step 8** Commit your changes or click **Dry Run** to check what will be configured on the devices before you commit.

**Step 9** Check that the new L3VPN service appears in the table and its provisioning state is **Success**.

# Step 4 Enable Service Health monitoring

After creating and provisioning the required L3VPN services, you can start monitoring their health.

### Before you begin

- (*Optional*) Ensure that Crosswork Service Health is installed. For details, see the Install Crosswork Applications chapter in Crosswork Network Controller 7.0 Installation Guide. For more information on Service Health, see Crosswork Network Controller 7.0 Service Health Guide.
- The Service Health related steps assume you have excess capacity available. Requirements (such as available resources, storage capacity, etc.) may be beyond the scope explained in this guide. See Crosswork Network Controller 7.0 Service Health Guide for details.

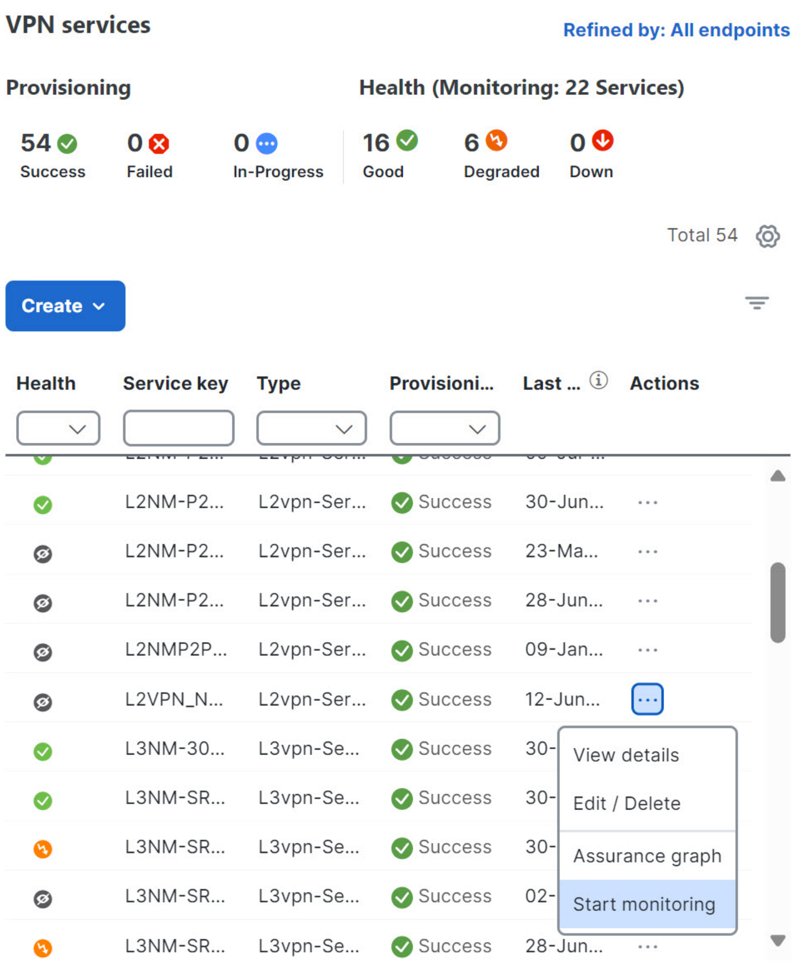
Select the newly created, unmonitored service, which will have a gray health indicator:

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The map opens on the left side of the page, and the table opens on the right side.

**Step 2** In the Actions column, click ⋯ for the service you want to start monitoring the health.

**Step 3** Click **Start Monitoring**.

*Figure 19: Start Monitoring VPN Services*

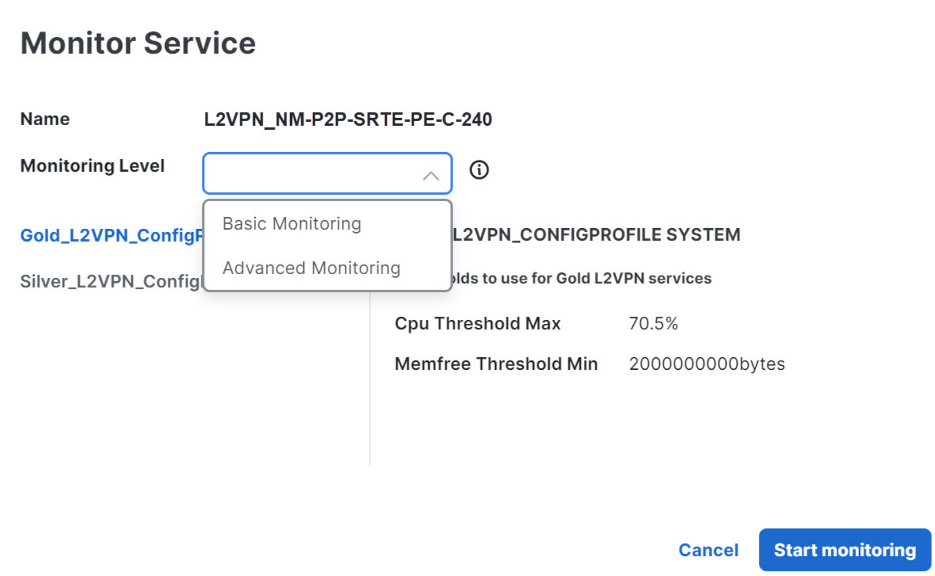



**Note** The Health column color coding indicates the health of the service:

- Blue = Initiated
- Green = Good
- Orange = Degraded
- Red = Down
- Gray = Not Monitoring

**Step 4** In the Monitor Service dialog box, select the Monitoring Level. For help with selecting the appropriate monitoring level for your needs, see Crosswork Network Controller 7.0 Service Health Guide.

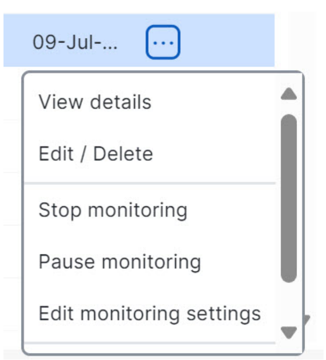
*Figure 20: Monitor Service*




**Step 5** Click **Start Monitoring**.

**Note** Once you have started monitoring the health of the service, in the Actions column, if you click ⋯ to view additional Service Health options, you will see: Stop Monitoring, Pause Monitoring, Edit Monitoring Settings, and Assurance Graph.

*Figure 21: Service Health Options*




**Step 6** Repeat these steps for each service that you wish to start health monitoring.

# Step 5 Visualize the new VPN service on the map to see the traffic path

**Step 1** In the L3VPN Service table, click on the service name or click ⋯ in the Actions column and choose **View Details** from the menu.

The map opens, and the service details are shown to the right of the map.

or

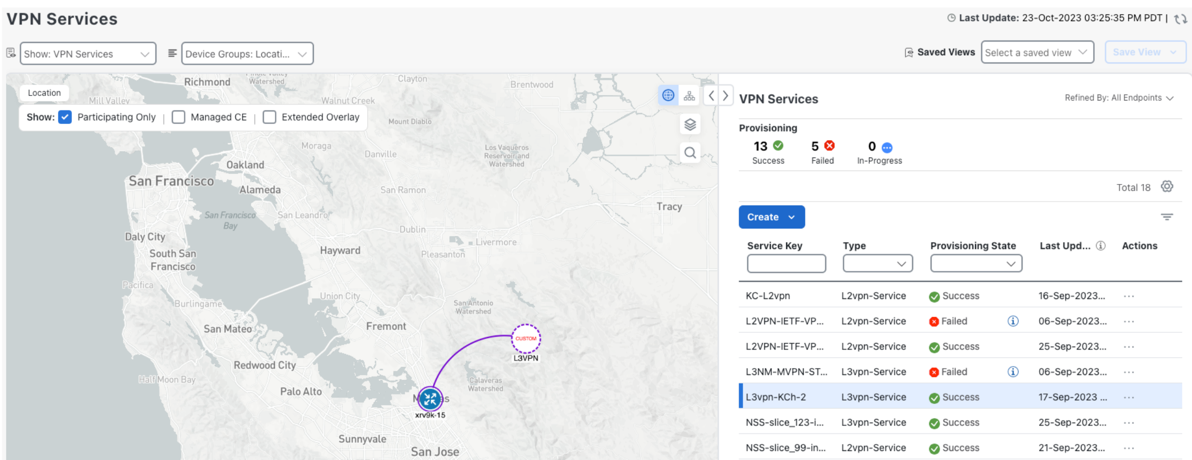
From the main menu, choose **Services & Traffic Engineering** > **VPN Services**.

The map opens, and a table of VPN services is displayed to the right of the map.

Click on the VPN in the Services table. If there are many services in the table, you can filter by name, type, or provisioning state to help locate the VPN.

In the map, the VPN is an overlay on the topology. It shows a representation of the three endpoints and a dashed line indicating that it is a virtual path.

***Figure 22: VPN service on the topology***




Select the **Show Participating Only** check box to see only the devices involved in the selected VPN.

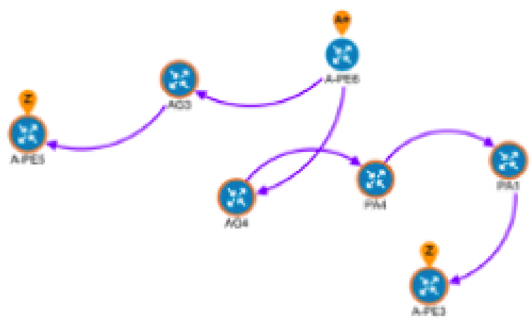
**Step 2** In the Actions column, click

to drill down to a detailed view of the VPN service, including the device configurations and the computed transport paths.

**Step 3** To see the computed paths for this VPN, click on the Transport tab in the Service Details pane. All the dynamically created SR-TE policies are listed in the Transport tab. Select one or more SR-TE policies to see the path from endpoint to endpoint on the map.

In this example, we examine the disjoint paths computed from PE-A to PE-B and from PE-A to PE-C.

*Figure 23: Computed disjoint paths*

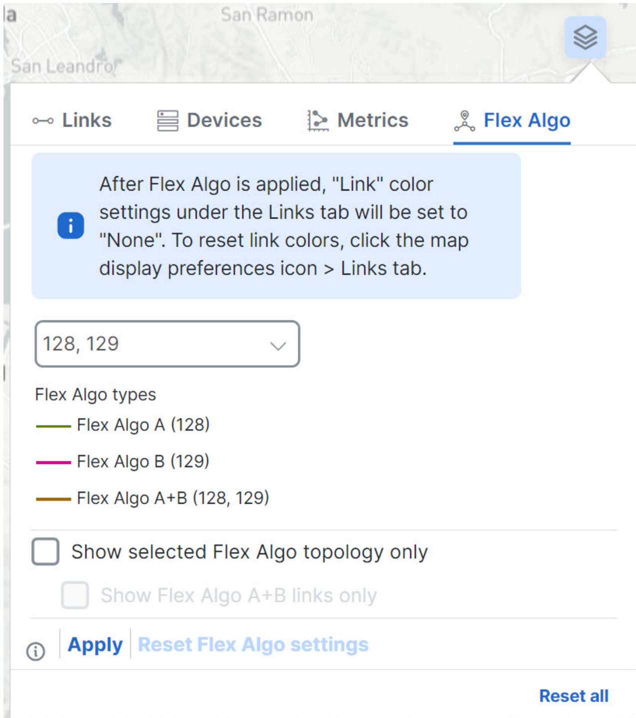


**Step 4** To see the physical path between the endpoints, select the **Show IGP Path** check box in the top-left corner of the map. Hover with your mouse over a selected policy in the table to highlight the path in the map and show prefix SID and routing information.

*Figure 24: Physical path details*

**Step 5** To filter the topology to a specific Flex-Algo constraint and visualize nodes and links you have configured manually in your network, click the button at the top right of the map and do the following:

*Figure 25: Flex-Algo tab*




a) Click the **Flex Algo** tab.
b) From the drop-down list, choose up to 2 Flex-Algo IDs.
c) View the Flex-Algo Types and confirm that the selection is correct. Also, note the color assignments for each Flex-Algo ID.
d) (Optional) Check the **Show selected Flex Algo topology only** check box to isolate the Flex-Algo IDs on the topology map. When this option is enabled, SR policy selection is disabled.
e) Check the **Show Flex Algo A+B links only** to show only those links and nodes that participate in both Flex-Algos.

   Suppose a selected Flexible Algorithm is defined with a criteria, but there are no links and node combinations that match it (for example, a defined affinity to include all nodes or links with the color blue). In that case, the topology map will be blank. If a selected Flexible Algorithm is not configured on a node or link, the default blue link or node color appears.

f) Click **Apply**. You must click **Apply** for any additional changes to your Flex-Algo selections to see the update on the topology map.
g) (Optional) Click **Save View** to save the topology view and Flexible Algorithm selections.

# Step 6 Observe automatic network optimization

### Observe automatic network optimization

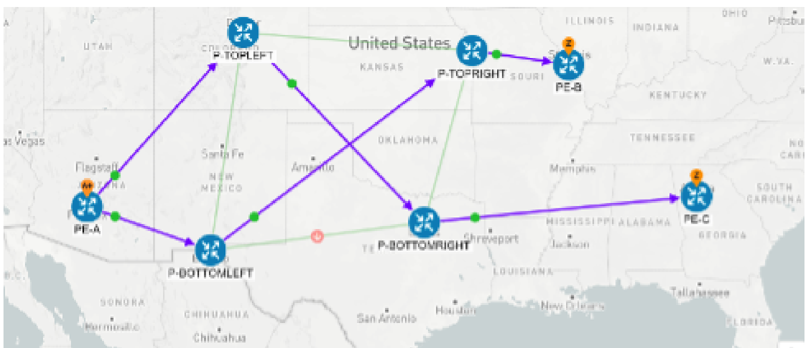
The SR-PCE constantly monitors the network and automatically optimizes the traffic path based on the defined SLA. For illustration purposes, let's look at what happens when one of the links goes down, in this case, the link between

P-BOTTOMLEFT and P-BOTTOMRIGHT. This means the previous path from PE-A to PE-C is no longer viable. Therefore, the SR-PCE computes an alternative path, from PE-A to PE-C and from PE-A to PE-B, to compensate for the down link and maintain the disjoint paths.

Recomputed paths:

| Source and Destination | Old path | New path |
|---|---|---|
| PE-A > PE-C | PE-A > P-BOTTOMLEFT > P-BOTTOMRIGHT > PE-C | PE-A > P-TOPLEFT > P-BOTTOMRIGHT > PE-C |
| PE-A > PE-B | PE-A > P-TOPLEFT > P-TOPRIGHT > PE-B | PE-A > P-BOTTOMLEFT > P-TOPRIGHT > PE-B |

*Figure 26: Automatic network optimization*




# Step 7 Inspect a degraded service using Service Health to determine active symptoms

By analyzing the root cause of reported active symptoms and impacted services, you can determine what issues must be addressed first to maintain a healthy setup and what requires further inspection and troubleshooting.

To view the active symptoms and root causes for service degradation:

### Before you begin

Ensure that service health monitoring is enabled for the service you want to inspect.

**Note** L3VPN service monitoring is supported on XR devices and not on XD devices. For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status remain in the degraded state with a METRIC_SCHEDULER error. To recover from this error, stop and restart the service monitoring.

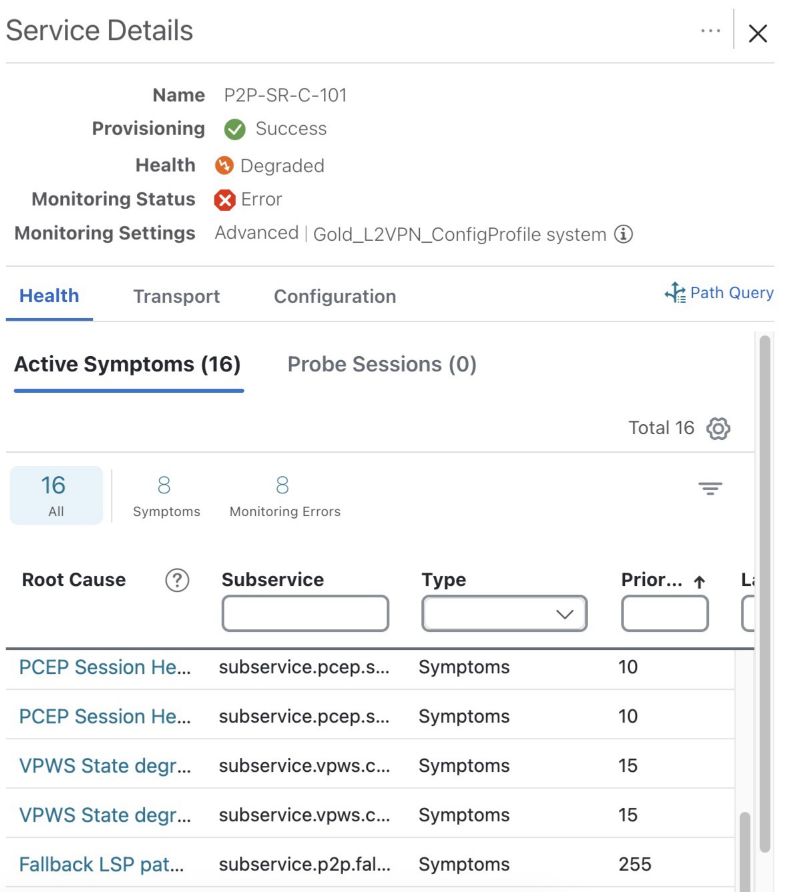
**Step 1** From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The map opens on the left side of the page, and the table opens on the right side.

**Step 2** In the Actions column, click ⋯ and click **View Details**. The Service Details panel appears on the right side.

**Step 3** Select the Health tab and click the **Active Symptoms** tab. The Active Symptoms table displays **Active Symptoms** and **Monitoring Errors** by default. To filter the table to show only the Active Symptoms, either click the **Symptoms** tab in the mini dashboard above the table or select **Symptoms** from the filter box under the **Type**. The table now shows a filtered list containing only the Active Symptoms.

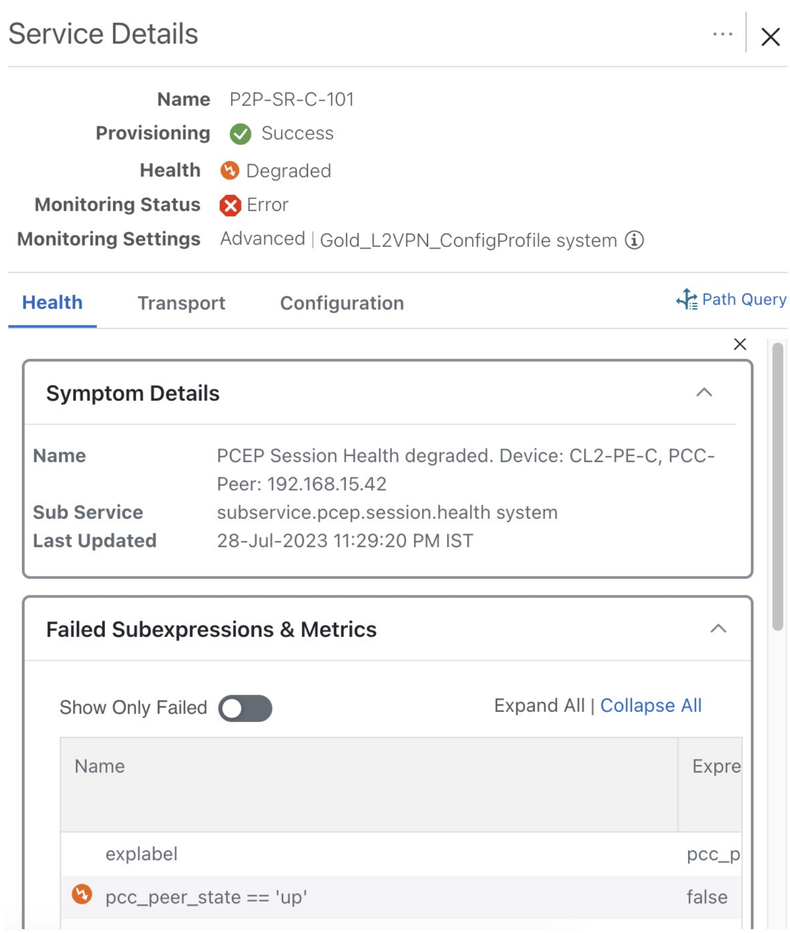
Review the Active Symptoms for the degraded service (including the Root Cause, Subservice, Type, Priority, and Last Updated details).

***Figure 27: Active Symptoms in Service Details***




**Step 4** Click on a Root Cause to view the **Symptom Details** and the **Failed Subexpressions & Metrics** information. As required, you can expand or collapse all of the symptoms listed in the tree. In addition, use the **Show Only Failed** toggle to focus only on the failed expression values.

*Figure 28: Root Cause in Service Details*




**Step 5** Click the **Transport** and **Configuration** tabs and review the details provided.

**Step 6** Click **X** in the top-right corner to return to the VPN Services list.

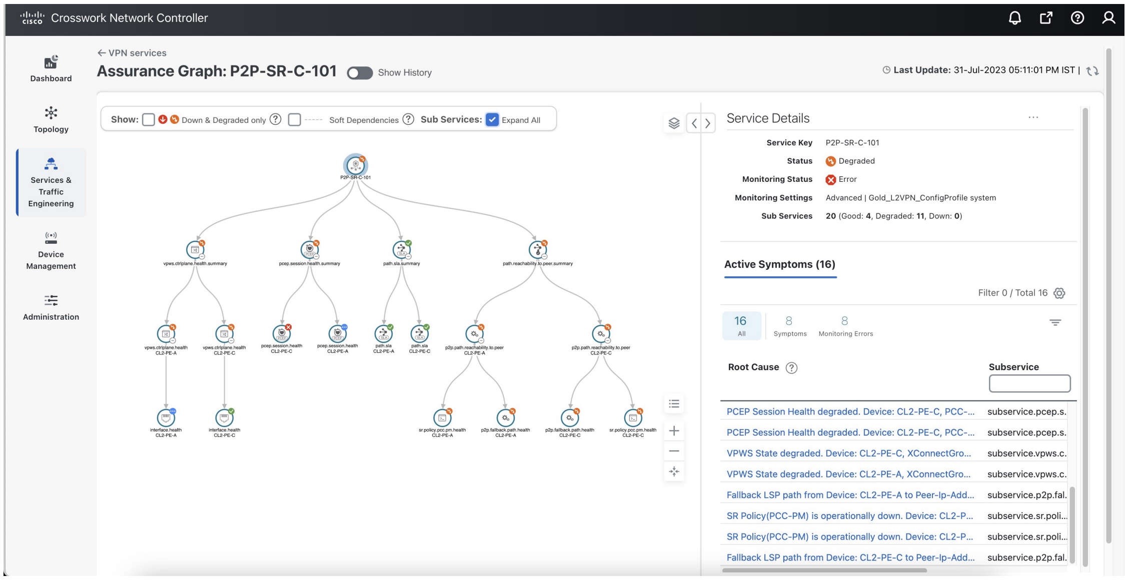
**Step 7** In the Actions column, click ⋯ for the required degraded service and click **Assurance Graph**. The topology map of services and subservices appear with the Service Details panel showing Service Key, Status, Monitoring Status, Monitoring Settings, Sub Services, and Active Symptoms details.

This may take up to 5-10 minutes to update after a service has been enabled for monitoring.

Metrics such as Jitter-RT (Jitter Round Trip), Latency-RT (Latency Round Trip), PktLoss-DS (Packet Loss from Destination to Source), and PktLoss-SD (Packet Loss from Source to Destination) also appear (information collected using Y.1731 probes). Additionally, a table of Active Symptoms listing Root Cause, Subservice, Priority, and Last Updated details is populated.

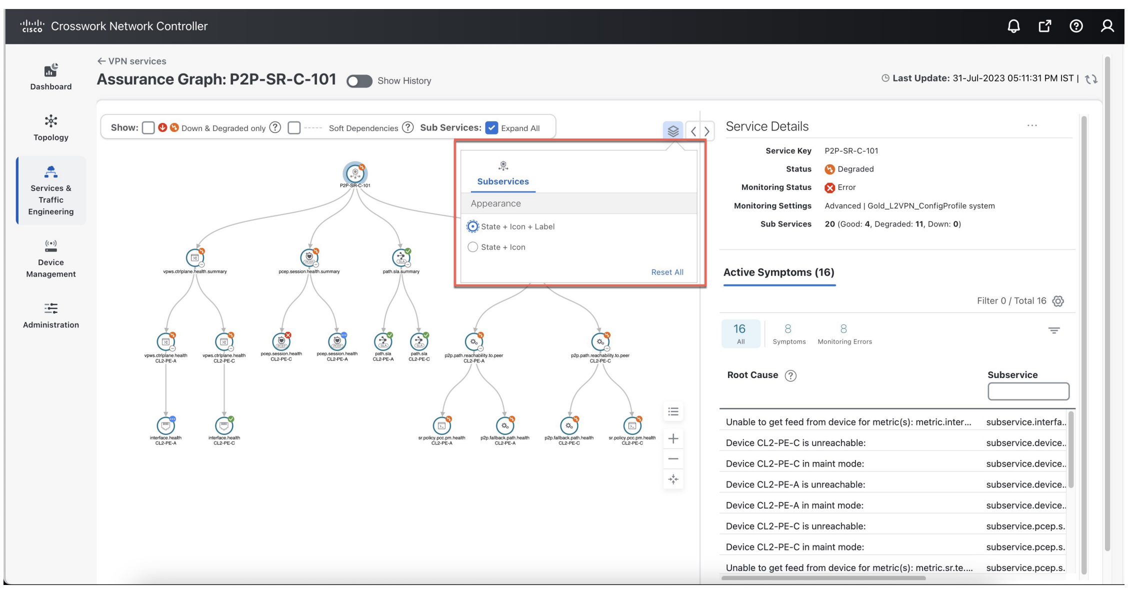
Step 7 Inspect a degraded service using Service Health to determine active symptoms

*Figure 29: Assurance Graph Details*




At the top-right of the map, select the stack icon to select the appearance option for the Subservices: **State + Icon + Label** or **State + Icon**.

*Figure 30: Appearance Option in Assurance Graph*



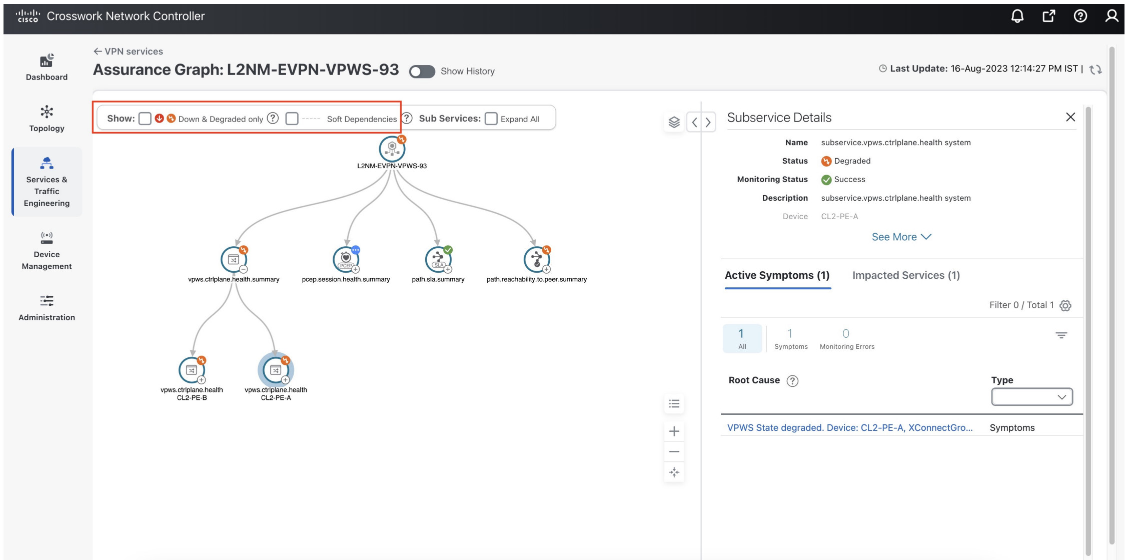

**Step 8** By default, the Assurance Graph displays a concise view with only the service and the top level subservices (aggregator nodes). Click the + icon in the nodes to expand the graph and to view the dependent details. To expand all the nodes at once, click the **Subservices: Expand All** check box at the top.

**Step 9** Select a degraded subservice in the Assurance Graph. The Subservice Details panel appears with subservice metrics, and subservice-specific Active Symptoms and Impacted Services details.

- **Active Symptoms**: Provides symptom details for nodes actively being monitored.

- **Impacted Services**: Provides information for services that are impacted by issues based on historical monitoring of health status.

***Figure 31: Degraded Subservice in the Assurance Graph***



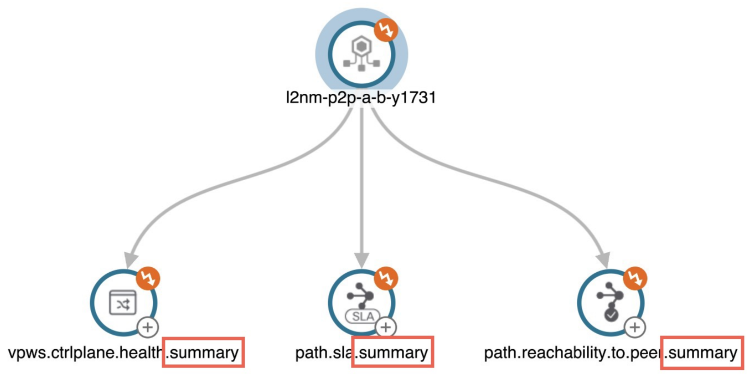

**Note** At the top left of the map, check the **Down & Degraded only** or **Soft Dependencies** check boxes to isolate the subservices further. Soft Dependencies imply that a child subservice's health has a weak correlation to its parent's health. As a result, the degraded health of the child will not result in the parent's health degradation.

**Note** In some cases, the Summary node feature is available and summarizes the aggregated health status of child subservices and reports a consolidated health status to a service node. The Summary node feature is available in both L2VPN multipoint Basic and Advanced monitoring models.

- Basic monitoring subservices:
  - Device—Summarizes the health status of all underlying Devices participating in the given L2VPN service.
  - Bridge Domain—Summarizes the L2VPN service's Bridge Domain health status across all participating devices.

- Advanced monitoring subservices (in addition to what is also available with Basic monitoring):
  - EVPN—Summarizes the health status of all underlying subservices, such as BGP Neighbor Health and MacLearning Health, across all participating PE endpoints and provides a consolidated overall EVPN health summary status.
  - Transport—Summarizes the health status of all underlying subservices—SR-ODN (dynamic), SR Policy (statically configured), and RSVP TE Tunnel, across all participating PE endpoints and provides a consolidated overall Transport health summary status.
  - SR-PCEP—Summarizes the health status of all the underlying subservices that are monitoring the PCEP sessions. Each underlying subservice monitors the health of the PCEP session on a particular device participating in the given VPN service.

*Figure 32: Summary Node Feature*




**Step 10** Inspect the Active Symptoms and Impacted Services information and the root causes associated with the degraded service to determine what issues need to be addressed to maintain a healthy setup.

To further troubleshoot a service health issue (such as a degraded device due to data fetching issues), follow these steps to determine whether the problem is related to a collection job.

**Step 11** From the main menu, choose **Administration** > **Collection Jobs**.

The Collection Jobs page appears.

**Step 12** Click the **Parameterized Jobs** tab.

**Step 13** Review the Parameterized Jobs list to identify the devices that may have service health degradation issues. By reviewing Parameterized Jobs, you can identify and focus on gNMI, SNMP, and CLI-based jobs by their Context ID (protocol) for further troubleshooting purposes.

**Step 14** In the Job Details panel, select the collection job you want to export and click to download the status of collection jobs for further examination. The information provided is collected in a .csv file when the export is initiated.

**Note** When exporting the collection status, you must fill in the information each time an export is executed. In addition, make sure to review the **Steps to Decrypt Exported File** content available on the Export Collection Status dialog box to ensure you can access and view the exported information.

**Step 15** Click **Export**.

**Step 16** To check the status of the exported collection job data, click **View Export Status** at the top right of the Job Details panel. The Export Status Jobs panel appears, providing the status of the export request.

**Step 17** Review the exported .csv file for collection job details and the possible cause of the degraded device.

# Summary and conclusion

As we observed in this example, operators can use Crosswork Network Controller to orchestrate L3VPNs with SLAs and to maintain these SLAs using SR-TE policies that continuously track network conditions and automatically react to optimize the network. This automation increases efficiency and reduces human error, which is generally unavoidable with manual tasks. Enabling Service Health to monitor provisioned services allows for more detailed symptoms, metrics, and analysis of each service.

# Scenario: Implement and maintain SLA for an L3VPN service for SRv6 (using ODN)

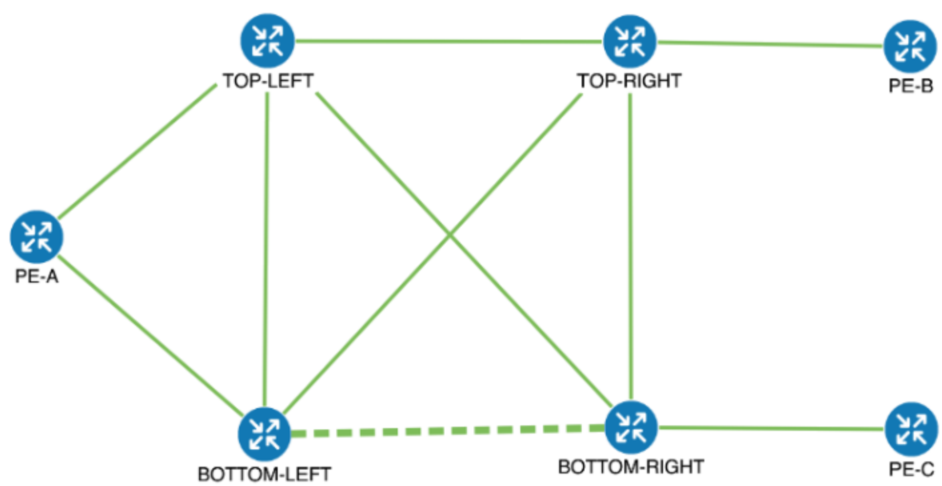
This scenario walks you through the procedure for provisioning an L3VPN service that requires a specific SLA objective. In this example, the lowest latency path is the SLA objective. The customer requires a low latency path for high-priority traffic. The customer wants to use disjoint paths, i.e., two unique paths that steer traffic from the same source to the same destination, avoiding common links so there is no single point of failure. The customer also wants to enable SRv6, which utilizes the IPv6 protocol to handle packets more efficiently, increase security and performance, and allow for a significantly larger number of possible addresses.

This is achieved using Segment Routing (SR) On-Demand Next Hop (ODN). ODN allows a service head-end router to automatically instantiate an SR-TE policy to a BGP next-hop when required (on-demand). The headend is configured with an ODN template with a specific color that defines the SLA upon which the traffic path will be optimized when a prefix with the specified color is received. Prefixes are defined in a route policy that is associated with the L3VPN.

Crosswork Network Controller continues to monitor the network and automatically optimizes it in a closed loop based on the defined SLA.

The following topology provides the base for this scenario:

***Figure 33: Example topology for the scenario***




In this scenario, we will:

- Create a segment routing ODN template with a specific color on the endpoints to ensure that traffic is transported within an LSP (underlay) and that a best-path tunnel is created dynamically when a prefix with the specified color is received. Enable SRv6 (IPv6) for service and link details. The ODN template defines the SLA on which you want to optimize the path. In this case, we will optimize on latency.
- Specify that the computed paths be disjoint: they will not share the same link.
- Create a route policy on each endpoint to be used to bind the L3VPN to the ODN template. This route policy adds a color attribute to the customer prefixes and advertises via BGP to other endpoints. This color attribute is used to indicate the SLA required for these prefixes.

- Create an L3VPN service with 3 endpoints: PE-A, PE-B, and PE-C. This is the overlay configuration.
- Visualize how this overlay/underlay configuration optimizes the traffic path and automatically maintains the SLA.

### Assumptions and prerequisites

- To use ODN with SRv6, BGP peering for the prefixes must be configured between the endpoints/PEs. Usually for L3VPN, this is the VPNv4 and VPNv6 address family peering, and this BGP peering must be over IPv6.

The procedure to implement and maintain SLA for an L3VPN service for SRv6 using ODN is detailed in this section.

## Step 1 Create an ODN template to map color to an SLA objective and constraints

We will create the following ODN templates:

- Headend PE-A, color 72, latency, disjoint path (link), group ID 16 - L3VPN_NM-SRTE-ODN_72-a
- Headend PE-A, color 71, latency, disjoint path (link), group ID 16 - L3VPN_NM-SRTE-ODN_71-a
- Headend PE-B and PE-C, color 70, latency - L3VPN_NM-SRTE-ODN_70
    - With multiple headends in the SRv6 enabled ODN template, the same locator name should be configured on the headend routers. Otherwise, different ODN templates should be created for each headend.
- Headend PE-B, color 72, latency - L3VPN_NM-SRTE-ODN_72-b
- Headend PE-C, color 71, latency - L3VPN_NM-SRTE-ODN_71-c

For example purposes, we will show how to create the first ODN template - L3VPN_NM-SRTE-ODN_72-a. The other ODN templates can be created using the same procedure.

### Before you begin

In this step, we will create an ODN template on each endpoint. The ODN template specifies the color and the intent; in this case, latency and disjointness. This ODN template will be used to dynamically create tunnels (on-demand) when prefixes with matching colors are received via BGP. Traffic to these prefixes will be automatically steered into the newly created tunnels, thereby meeting the SLA objective and constraints intended for these prefixes and signaled using colors in the BGP routes.

Disjointness constraints work by associating a disjoint group ID with the ODN template. Tunnels with the same disjoint group ID will be disjoint, i.e., they will use different links, nodes, and shared risk link groups depending on how the disjoint groups are configured.

---

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **SR-TE** > **ODN-Template**.

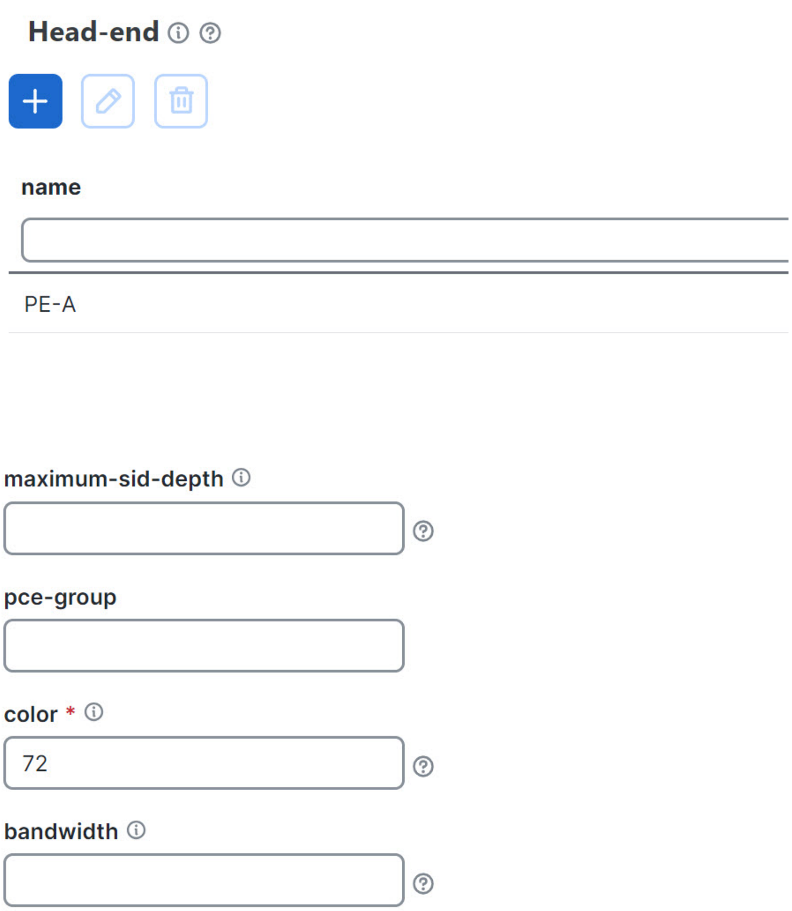
**Step 2** Click + to create a new template and give it a unique name.

In this case, the name is **L3VPN_NM-SRTE-ODN_72-a**. Click **Continue**.

*Figure 34: SR-TE ODN-Template*


**Step 3** Choose the headend device, **PE-A**, and specify the color **72**.

*Figure 35: ODN-Template Head-end Routers*

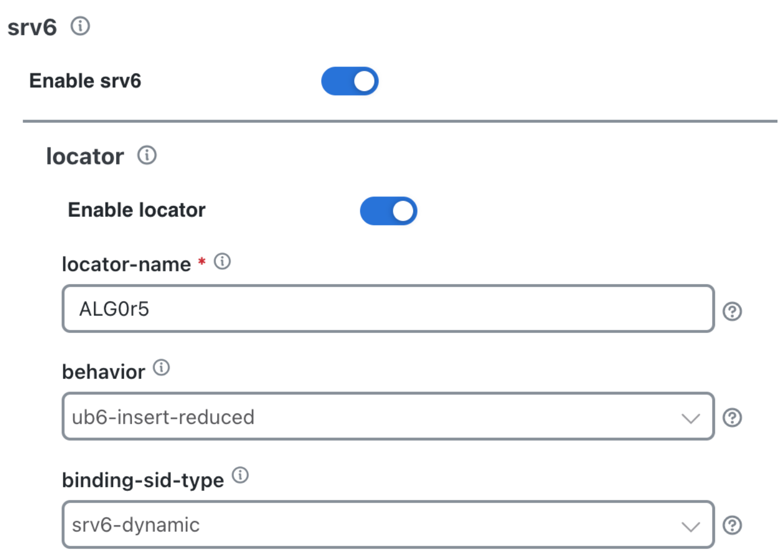



**Step 4** Under srv6, select the **Enable srv6** toggle.

**Step 5** Under locator, enter the required SRv6 **locator-name**.

The locator name should match what is configured on the router.

*Figure 36: ODN-Template SRv6 options*



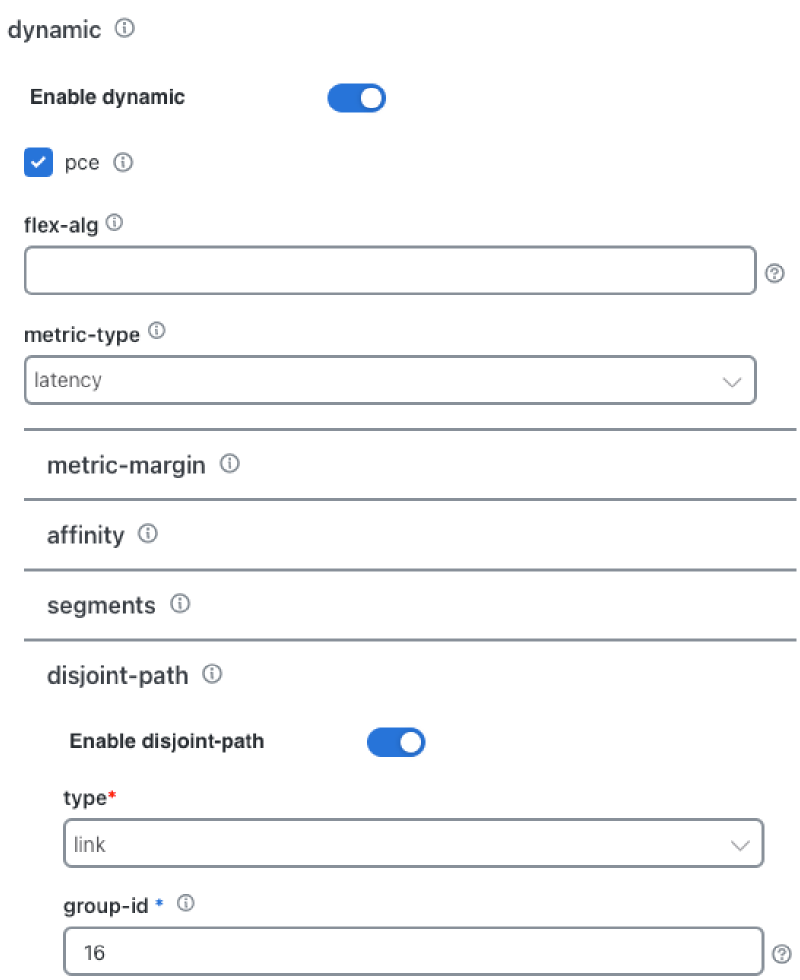

**Step 6** Under dynamic, select **latency** as the metric type. This is the SLA objective on which we are optimizing.

**Step 7** Select the **pce** check box to specify that the path should be computed by the SR-PCE, not by the Path Computation Client (PCC).

**Step 8** Define the required constraints. In this case, we want the computed paths to be disjoint in that they must not share a link.

Under disjoint-path, choose **link** as the type, and specify a numeric group ID, in this case, 16.

*Figure 37: ODN-Template Dynamic Path Computation*




Commit your changes or click **Dry Run** to check what will be configured on the devices before you commit.

**Step 9** Check that the new ODN template appears in the table and its provisioning state is **Success**. Click in the Actions column and choose **Config View** to see the Yang model-based service intent data that details the ODN template you created.

*Figure 38: ODN Template Actions Menu*

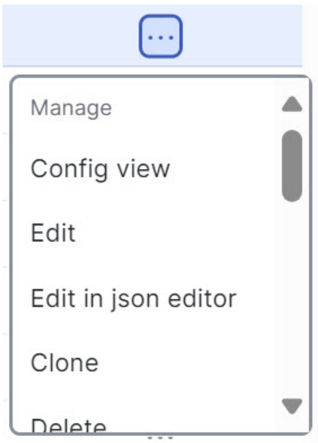
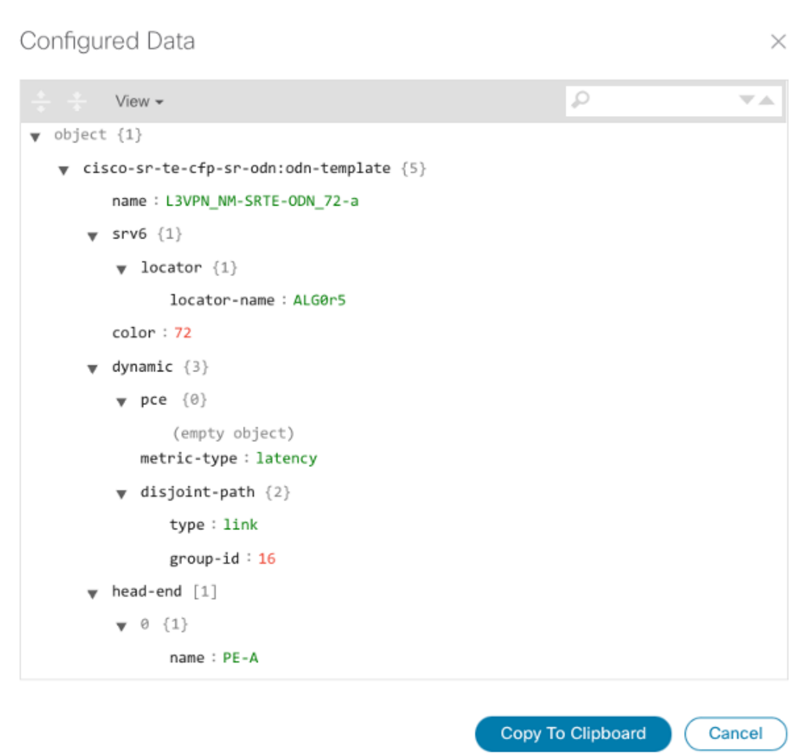



*Figure 39: Configured Data*




**Step 10** Create the other ODN templates listed above in the same manner.

# Step 2 Create an L3VPN route policy

In this step, we will create a route policy for each endpoint, and we will specify the same color as defined in the ODN template for that endpoint. The route policy defines the prefixes to which the SLA applies. When traffic from the specified network with a matching color is received, paths are computed based on the SLA defined in the ODN template. We will create the following route policies:

- Color 70, IPv6 prefix 70:70:70::0/64 - L3VPN_NM-SRTE-RP-PE-A-7
- Color 71, IPv6 prefix 70:70:71::0/64 - L3VPN_NM-SRTE-RP-PE-B-7
- Color 72, IPv6 prefix 70:70:72::0/64 - L3VPN_NM-SRTE-RP-PE-C-7

For example purposes, we will show how to create the first route policy - L3VPN_NM-SRTE-RP-PE-A-7. The other route policies can be created using the same procedure.

First, we will create the routing policy tag and routing policy destination prefix. The routing policy prefixes should match with the subnet prefix configured on the PE devices in the service.

---

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **Routing Policy Tag**.

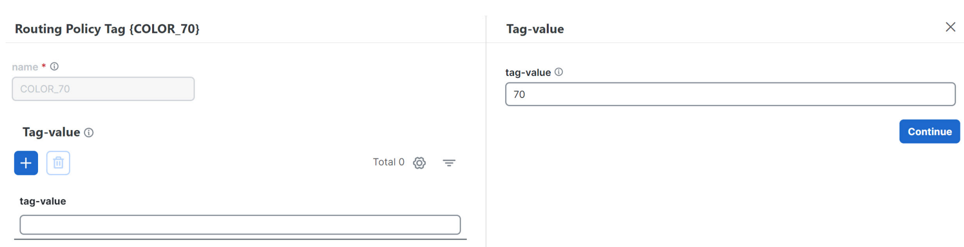
**Step 2** Click + to create a new routing policy tag and type the name of the tag set: **COLOR_70**. Click **Continue**.

This is used as a label to reference the set in actions and conditions.

**Step 3** Under tag-value, click + and type the Tag-value: **70**.

The tag value may be a number between **1 – 4294967295** and should match to a color value.

*Figure 40: Routing policy tag*




**Step 4** Click **Continue**. The new routing policy tag name with the new tag value is visible. Click **Commit changes**.

Create the other two routing policy tags (**COLOR_71** and **COLOR_72**) and tag values (**71** and **72**) by following the same steps above.

Now, create the routing policy destination prefixes.

**Step 5** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **Routing Policy Destination Prefix**.

**Step 6** Click + to create a new routing policy destination prefix and type the name **DEST_PREFIX_SET_70**.

The name of the prefix set will reference the set in match conditions.

**Step 7** For Mode, select **ipv6**.

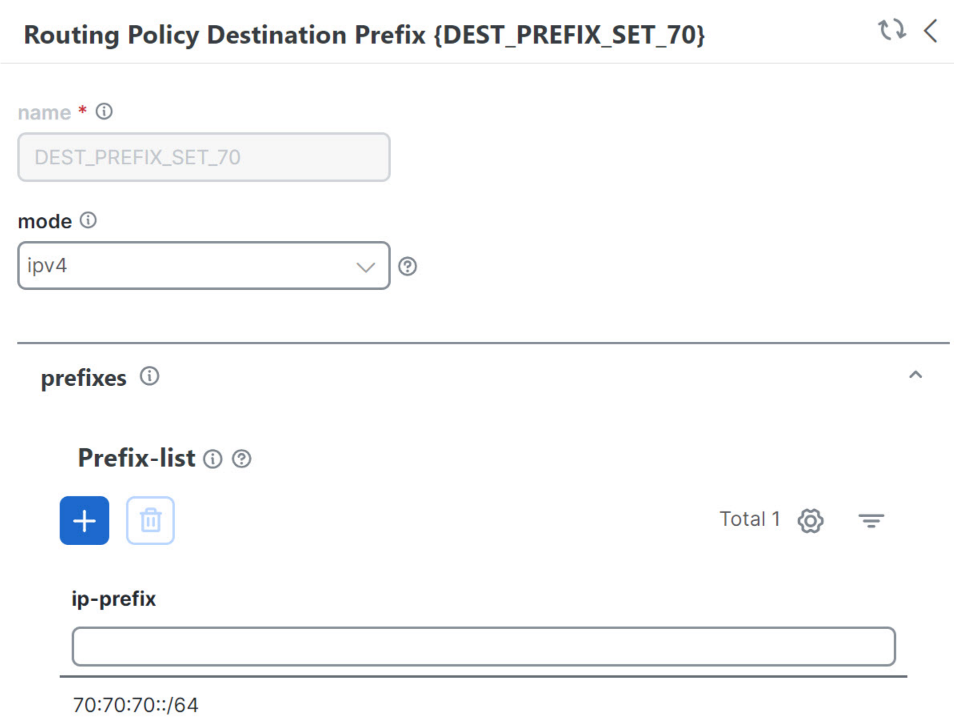
**Step 8** Expand prefixes and click + to add the ip-prefix to the prefix-list.

**Step 9** For Ip-prefix, type **70:70:70::0/64** and click **Continue**.

Create the other two routing policy destination prefixes (**DEST_PREFIX_SET_71** and **DEST_PREFIX_SET_72**) by following the same steps. Click **Commit changes**.

Now, we are ready to create the first route policy, L3VPN_NM-SRTE-RP-PE-A-7. The other route policies can be created using the same procedure.

***Figure 41: Routing policy destination prefix***




**Step 10** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **Routing Policy**.

**Step 11** Click + to create a new route policy and type a unique name for the top-level policy definition: **L3VPN_NM-SRTE-RP-PE-A-7**. Click **Continue**. The statements section appears.

**Note** The Route Policy statement defines the condition and actions taken by the system.

**Step 12** Expand statements, click + to add the name of the policy statement (such as **stmt1**), and click **Continue**. The statement{stmt1} panel appears, showing **conditions** and **actions** sections.

**Step 13** Expand conditions and then expand match-dest-prefix-set before selecting the prefix-set list and select **DEST_PREFIX_SET_70**. This is what references a defined prefix set.

**Note** Once selected, the **Enable match-dest-prefix-set** toggle switches on. This toggle will match a referenced prefix set according to the logic defined in the match-set-options list.

**Step 14** Expand actions and then expand bgp-actions.

**Step 15** For bgp-actions, slide the Enable bgp-actions toggle to the on position. By toggling bgp-actions on, it defines the top-level container for BGP-specific actions.

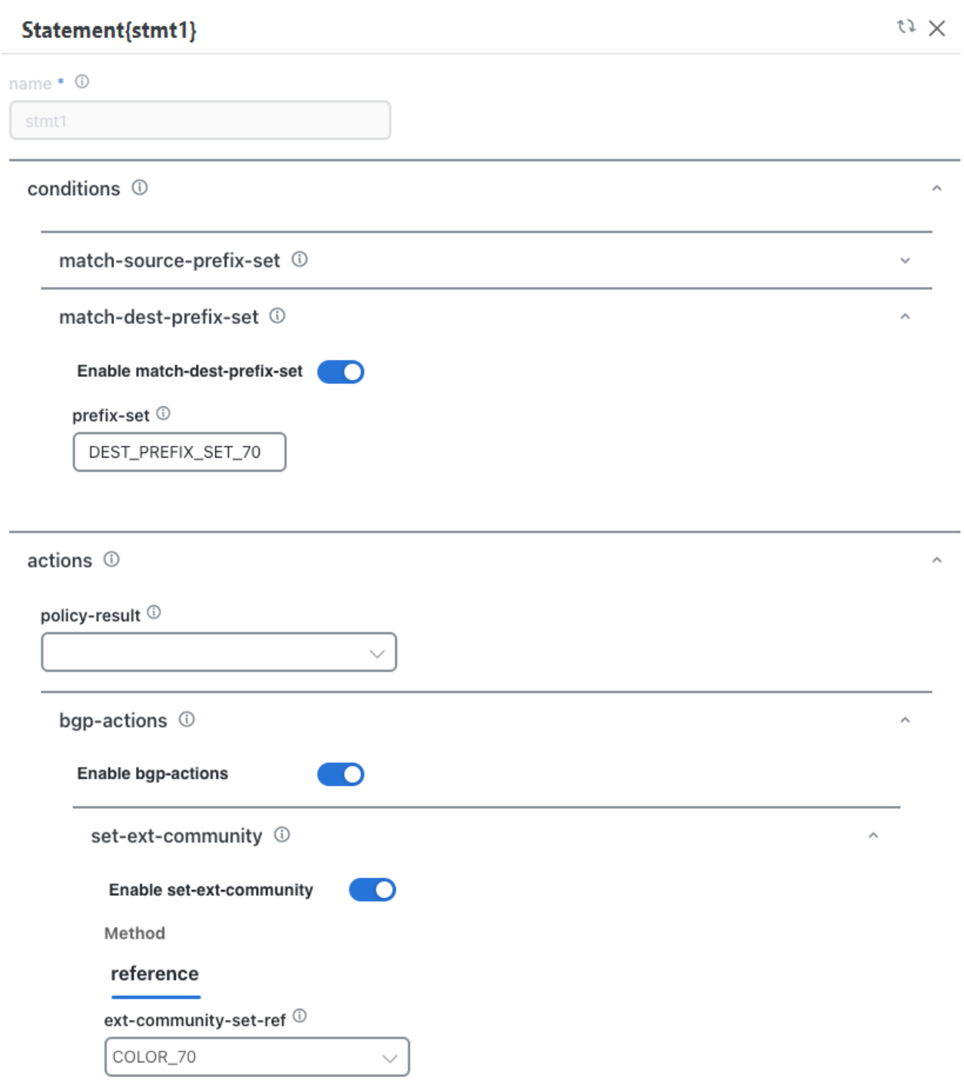
**Step 16** Now expand set-ext-community. Slide the enable-set-ext-community toggle to the on position. Toggling on set-ext-community sets the extended community attributes.

**Step 17** For Method and reference, select the ext-community-set-ref list and select **COLOR_70**. The Ext-community-set-ref references a defined extended community set by name.

**Note** Creating routing-policy tag-set is mandatory and needs to be mapped here.

**Step 18** Click **X** in the top-right corner to close the statement{stmnt1} panel and click **Commit changes**.

*Figure 42: Routing policy statement*




**Step 19** Create the other route policies (L3VPN_NM-SRTE-RP-PE-B-7 and L3VPN_NM-SRTE-RP-PE-C-7) in the same manner.

After creating the L3VPN route policies, create the VPN profile for each route policy and then create and provision the L3VPN service. The VPN profile will be referenced from the L3VPN service. This will bind the route policy to the L3VPN service.

# Step 3 Create and provision the L3VPN service

In this step, we will create the L3VPN service with three endpoints: PE-A, PE-B, and PE-C. Each endpoint will be associated with a vpn-instance-profile, which in turn points to a VPN profile that contains the route policy with the same color as specified in the ODN template. In this way, traffic that matches the specified prefixes and color will be treated according to the SLA specifications.

First, we will create the VPN profiles. The newly created VPN profiles will have the same names as the L3VPN routing policy names.

---

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3VPN** > **VPN Profiles**.

**Step 2** Click + to create a valid VPN profile to be referenced in the VPN service.

**Step 3** Select the Id list and select **L3VPN_NM-SRTE-RP-PE-A-7**.

Now create and provision the L3VPN service.

**Step 4** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L3vpn** > **L3vpn-Service**..

**Step 5** Click + to create a new service and type a new vpn-id: **L3VPN_NM-SRTE-ODN-70**.

A VPN identifier uniquely identifies a VPN and has a local meaning (for example, within a service provider network).

**Step 6** Click **Continue**.

**Step 7** Create vpn-instance-profiles, which is a container that defines the route distinguisher (RD), route targets, and the export/import route policy. We will create vpn-instance-profiles for each endpoint, as follows:

- L3VPN_NM_SR_ODN-IE-PE-A-7 with route distinguisher 0:70:70
- L3VPN_NM_SR_ODN-IE-PE-B-7 with route distinguisher 0:70:71
- L3VPN_NM_SR_ODN-IE-PE-C-7 with route distinguisher 0:70:72

**a.** Expand vpn-instance-profiles and click + to create a new vpn-instance-profile profile-id: **L3VPN_NM_SR_ODN-I-PE-A-7**. Click **Continue**.

**b.** Enter the route distinguisher (Rd) that will differentiate the IP prefixes and make them unique: **0:70:70**.

**c.** For address-family, click + and select **ipv6** from the list. Click **Continue**.

**d.** Define the required VPN targets, including route targets and route target types (import/export/both).

**e.** Under vpn-policies, in the Export-policy list, choose the relevant VPN profile (which contains the route policy: **L3VPN_NM-SRTE-RP-PE-A-7**). This forms the association between the VPN and the ODN template that defines the SLA.

**f.** Click **X** in the top-right corner when you are done.

**g.** Expand srv6, slide the Enable srv6 toggle to the on position, and then click + under address-family.

**h.** Select **ipv6** from the address family list and click **Continue**.

**i.** For Locator-name, type **ALG0r5**. The SRv6 locator name should match locators configured at a node-global level on each router. Click **X** in the top-right corner until you are back on the Create L3VPN screen.

**j.** Similarly, create the other vpn-instance-profiles.

**Step 8** Define each VPN endpoint individually: PE-A, PE-B, and PE-C.

a) Expand vpn-nodes and click + to select the relevant device from the list: **PE-A**. Click **Continue**.

b) Enter the local autonomous system number for network identification: **200**.

c) Expand active-vpn-instance-profiles and click + to select the Profile-id you created in the previously: **L3VPN_NM-SRTE-RP-PE-A-7**. Click **Continue**.

d) Define the network access parameters for communication from the PE to the CE:

- Under vpn-network-accesses, click + to create a new set of VPN access parameters and provide a unique ID. Click **Continue**.
- In the Interface-id field, type **Loopback70**. This is the identifier for the physical or logical interface. The identification of the sub-interface is provided at the connection level and/or the IP connection level.
- Expand ip-connection > ipv6 and enter a Local-address (**70:70:70::1**) and the Prefix-length (**64**).
- Expand routing-protocols and click + before typing a unique identifier for the routing protocol: **EBGP**. Click **Continue**.
- From the routing protocol Type list, select **bgp-routing**.
- Expand bgp and for Peer-as, type **70**. This information indicates the customer's ASN when the customer requests BGP routing.
- From the Address-family list, select **ipv6**.
- Under neighbor, click + to create a neighbor IP address and type **70:70:70::2**. Click **Continue**.
- Type the Multihop number **11**. This describes the number of IP hops allowed between a given BGP neighbor and the PE.
- For redistribute-connected, click + and select **ipv6** from the Address-family list. Click **Continue**.
- Click **X** in the top-right corner until you are back on the Create L3VPN screen.
- Similarly, create the other VPN nodes: **PE-B** and **PE-C**.

**Step 9** Commit your changes or click **Dry Run** to check what will be configured on the devices before you commit.

**Step 10** Check that the new L3VPN service appears in the table and its provisioning state is **Success**.

## Step 4 Visualize the new VPN service on the map to see the traffic path

**Step 1** In the L3VPN Service table, click on the service name or click ... in the Actions column and choose **View Details** from the menu.

The map opens, and the service details are shown to the right of the map.

or

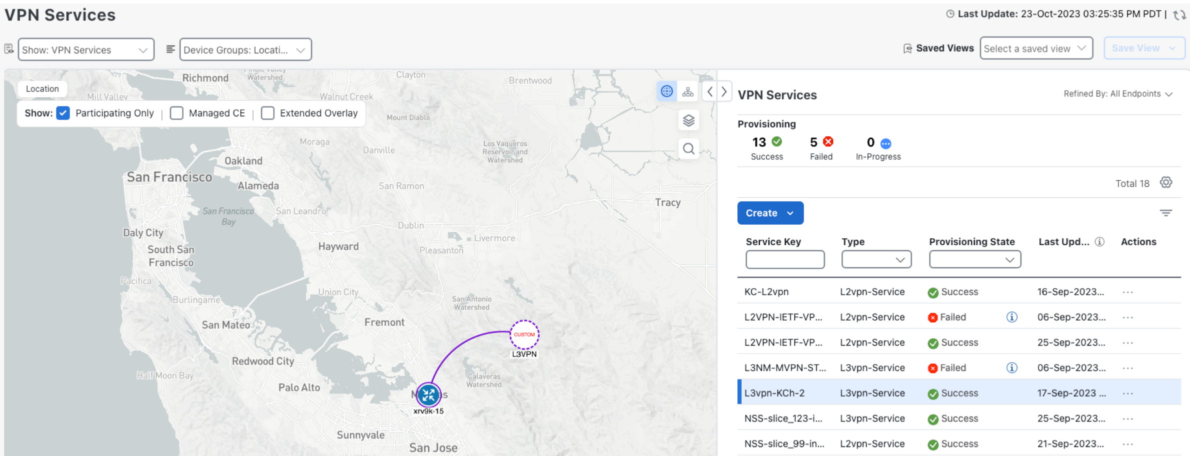
a) From the main menu, choose **Services & Traffic Engineering** > **VPN Services**.

The map opens, and a table of VPN services is displayed to the right of the map.

b) Click on the VPN in the Services table. If there are many services in the table, you can filter by name, type, or provisioning state to help locate the VPN.

In the map, you will see the VPN as an overlay on the topology. It shows a representation of the three endpoints and a dashed line that indicates that it is a virtual path.

*Figure 43: VPN service on the topology*




Select the **Show Participating Only** check box to see only the devices involved in the selected VPN.

**Note** When a Provision State shows a Failed state, an information icon appears. This is true whether you are on the VPN Services, Service Details, and many of the Provisioning screens that show a table of services and their Provisioning status. If you select the icon, Error Message details appear describing the failure. You can also click the **Show Error Details** link to view the Component Errors screen and take action to fix the error. Each failed source provides further error message details and recommendations. For example, in the Action column for the failed source on the component Errors screen, you may click ⋯ for different options (such as **Check-Sync**, **Sync-To**, **Sync-From**, **Compare-Config**, **View Job Status**) that will assist in fixing the error. Service level actions are also available for additional options (such as **Re-Deploy**, **Reactive-Re-deploy**, **Re-Deploy Reconcile**, **Clean-up**, etc.) that will assist in fixing the service level error. Use the information icons next to these options for further details on fixing the issue.

*Figure 44: Provisioning status information icon*

*Figure 45: Error message*

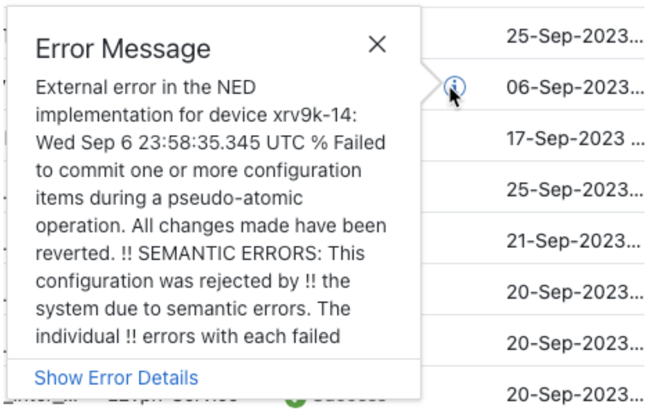



*Figure 46: Component errors action menu*

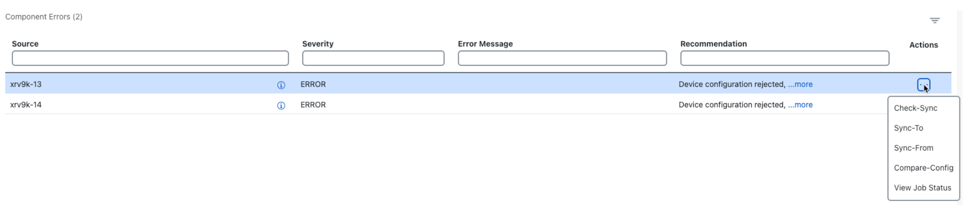
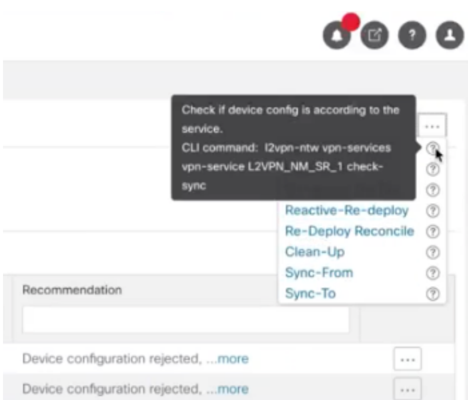
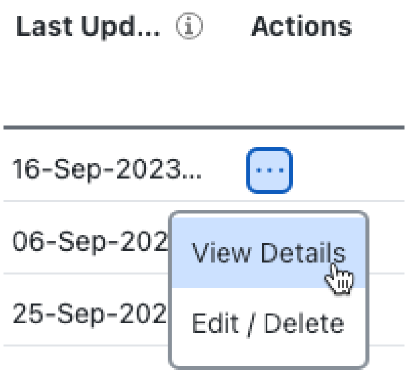



*Figure 47: Service level actions*




**Step 2** In the Actions column, click ⋯ to drill down to a detailed view of the VPN service, including the device configurations and the computed transport paths.

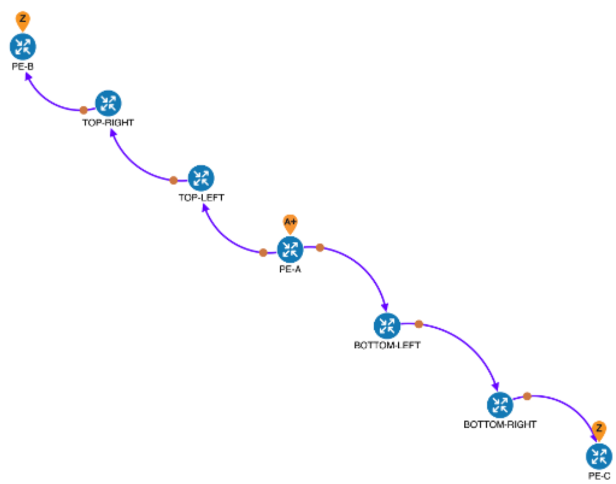
*Figure 48: View details in actions menu*




**Step 3** To see the computed paths for this VPN, click on the Transport tab in the Service Details pane. All the dynamically created SR-TE policies are listed in the Transport tab. Select one or more SR-TE policies to see the path from endpoint to endpoint on the map.

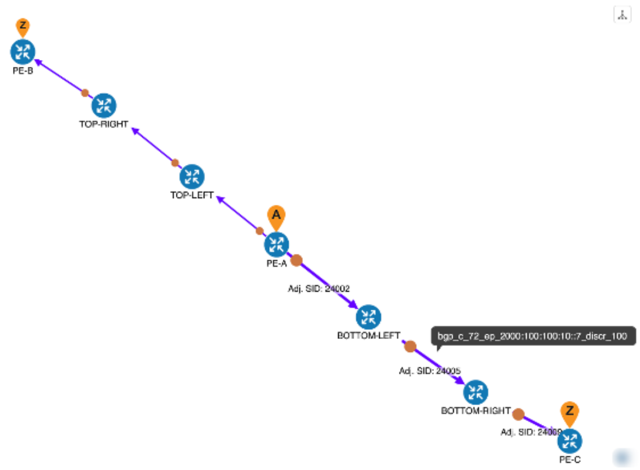
In this example, we are looking at the disjoint paths computed from PE-A to PE-B and from PE-A to PE-C.

*Figure 49: Computed disjoint paths*




**Step 4** To see the physical path between the endpoints, select the **Show IGP Path** check box in the top-left corner of the map. Hover with your mouse over a selected policy in the table to highlight the path in the map and show prefix SID and routing information.

*Figure 50: Physical path details*



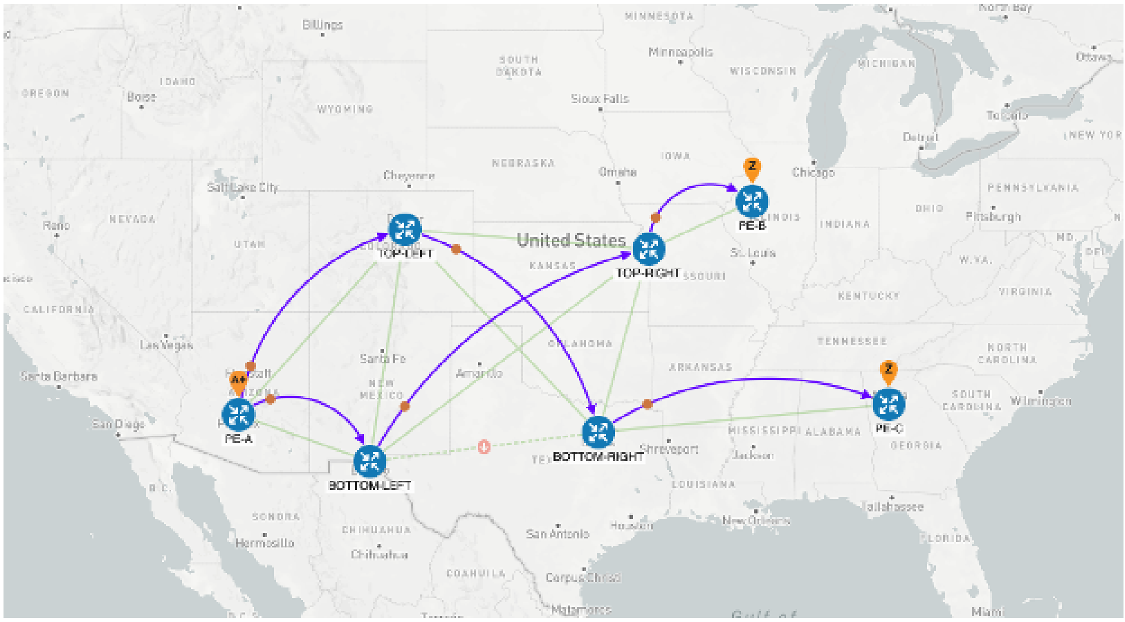

# Step 5 Observe automatic network optimization

The SR-PCE constantly monitors the network and automatically optimizes the traffic path based on the defined SLA. For illustration purposes, let's look at what happens when one of the links goes down, in this case, the link between P-BOTTOMLEFT and P-BOTTOMRIGHT. This means the previous path from PE-A to PE-C is no longer viable. Therefore, the SR-PCE computes an alternative path, both from PE-A to PE-C and from PE-A to PE-B, to compensate for the link that is down and maintain the disjoint paths.

Recomputed paths:

| Source and Destination | Old path | New path |
|---|---|---|
| PE-A > PE-C | PE-A > BOTTOM-LEFT > BOTTOM-RIGHT > PE-C | PE-A > TOP-LEFT > BOTTOM-RIGHT > PE-C |
| PE-A > PE-B | PE-A > TOP-LEFT > TOP-RIGHT > PE-B | PE-A > BOTTOM-LEFT > TOP-RIGHT > PE-B |

*Figure 51: Automatic Network Optimization*




## Summary and Conclusion

As we observed in this example, operators can use Crosswork Network Controller to orchestrate L3VPNs for SRv6 with SLAs and maintain these SLAs using SR-TE policies that continuously track network conditions and automatically react to optimize the network. This automation increases efficiency and reduces human error, which is generally unavoidable with manual tasks.

# Scenario: Mandate a static path for an EVPN-VPWS service using an explicit MPLS SR-TE policy

To ensure that mission-critical traffic within a VPN traverses the higher-capacity interfaces rather than the lower-capacity interfaces, we will create a point-to-point EVPN-VPWS service and associate a preferred path (explicit) MPLS SR-TE policy on both endpoints for service instantiation. In this way, we will mandate a static path for the mission-critical traffic.

In this scenario, we will see how quick and easy it is to create SR-TE policies and VPN services by uploading a file containing all the required configurations. We will download sample files (templates) from the provisioning UI, fill in the required data, and then import the file via the UI. Lastly, we will use the Service Health functionality to review the health of the services and view the Assurance Graph and Last 24Hr Metrics to better analyze our service's health details.

**Note** In this scenario, reference to SR-TE specifically means SR-TE over MPLS.

In this scenario, we will:

- Create an SID list - a list of prefix or adjacency Segment IDs, each representing a device or link along the path.
- Provision an explicit SR-TE policy, which will reference the SID list, thus creating a predefined path into which the EVPN prefix will be routed.
- Provision a point-to-point EVPN-VPWS service from PE-A to PE-C and attach the explicit SR-TE policy.
- Visualize the path of the service and review its health.

### Assumptions and prerequisites

- For transport mapping to L2VPN service, devices must be configured with the **l2vpn all** command.
- For Service Health enablement, Service Health must be installed. For instructions, see Crosswork Network Controller Installation Guide chapter, Install Crosswork Applications.
- (Optional) Service Health provides **Internal Storage** of monitoring data up to a maximum limit of 50 GB. This data is stored on your system. IIf you exceed the internal storage limit, historical data will be lost. If you extend Service Health storage capacity, you can configure **External Storage** in the cloud using an Amazon Web Services (AWS) cloud account. By leveraging External Storage, all existing internal storage data will be automatically moved to the external cloud storage (see **Configuring Service Health External Storage Settings** for more details) and your internal storage will act locally as cache storage. Configuring External Storage for Service Health ensures you will not lose historical data for services that continue to monitor a service's health and will retain service health data for any service you choose to stop monitoring when you select the option to retain historical monitoring service for the data. For more information on Internal and External Storage and how to retain historical monitoring service data when stopped, see Crosswork Network Controller 7.0 Service Health Guide.
- Before using Service Health's Assurance Graph, ensure that topology map nodes have been fully configured and created with a profile associated with the service. If not, Subservice Details metrics will show that no value has yet to be reported.
- For Service Health, you must configure two buckets on the Y1731 profile associated with the device. If you have fewer than two buckets configured, Service Health cannot report the Y1731 probes/KPIs on the service details page.

# Step 1 Prepare for creating a SID list

### Before you begin

The SID list consists of a series of prefixes or adjacency SIDs, each representing a node or link along the path. Each segment is an end-to-end path from the source to the destination, and it instructs the routers in the network to follow the specified path instead of the shortest path calculated by the IGP.

To build the SID list, you will need the MPLS labels of the desired traversing path. You can get these labels from the devices themselves, or you can invoke the northbound Cisco Crosswork Optimization Engine API to retrieve this information.

Refer to Cisco Crosswork Network Automation API Documentation on Cisco Devnet for more information about the API.

**Step 1** Prepare the input required to produce the SID list for the path from endpoint to endpoint. You will need the router ID of each endpoint, as follows:

```
{
  "input": {
    "head-end": "100.100.100.7",
    "end-point": "100.100.100.5",
    "sr-policy-path": {
      "path-optimization-objective": "igp-metric"
    }
  }
}'
```

**Step 2** Invoke the API on Crosswork Network Controller server by using the input prepared in the previous step. For example:

```
curl --location --request POST
'https://10.194.63.198:30603/crosswork/nbi/optima/v1/restconf/operations/cisco-crosswork-
optimization-engine-sr-policy-operations:sr-policy-dryrun' \
--header 'Content-Type: application/yang-data+json' \
--header 'Accept: application/yang-data+json' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsImlzRnJvbU5ld0xvZ2luIjoidHJ1ZSIsInBvbGljeV9pZCI6ImFkb
WluIiwiYXV0aGVudGljYXRpb25EYXRlIjoiMjAyMS0wMy0yMlQxNjozODozNy43NDY2MTZaW0dNVF0iLCJzdWNjZXNzZnV
sQXV0aGVudGljYXRpb25IYW5kbGVycyI6IlF1ZXJ5RGF0YWJhc2VBdXRoZW50aWNhdGlvbkhhbmRsZXIiLCJpc3MiOiJod
HRwOlwvXC9sb2NhbGhvc3Q6NTQ4OVwvU1NPIiwibGFzdF9uYW1lIjoic21pdGgiLCJjcmVkZW50aWFsVHlwZSI6IlVzZXJ
uYW1lUGFzc3dvcmRDcmVkZW50aWFsIiwiYXVkIjoiaHR0cHM6XC9cLzEwLjE5NC42My4xOTg6MzA2MDNcL2FwcC1kYXNoY
m9hcmQiLCJhdXRoZW50aWNhdGlvbk1ldGhvZCI6IlF1ZXJ5RGF0YWJhc2VBdXRoZW50aWNhdGlvbkhhbmRsZXIiLCJsb25
nVGVybUF1dGhlbnRpY2F0aW9uUmVxdWVzdFRva2VuVXNlZCI6ImZhbHNlIiwiY2hhbmdlX3B3ZCI6ImZhbHNlIiwiZXhwI
joxNjE2NDU5OTIwLCJpYXQiOjE2MTY0MzExMjAsImZpcnN0X25hbWUiOiJqb2huIiwianRpIjoiU1QtODQtOFVlWXMybEt
3R2d1Z3RIYjl4MzVmTFlNTGVVRlp6OURyNGpoeFcxakhsV01VYXdXSWgxbUdTd01aRC1tOEk1S2Z0amI2ZmlWTUhlYnBYY
jBMMFZqRFc2WVppUFVUbHRpNFVpZnNUeG9aQ284WWpPWEc2VlFjS0Mwb29lWjJhc3BWanMzYnA3bHo5VkhySlBCTzl5TDN
GcFRIWXRPeWJtVi1jYXMtMSJ9.Vi4k0w8KsOv5M_O8zBqWochT3k9V9Pn2NjSn5ES9c5Pf-
4ds0o4kk6xuZx5_cggauiEICuUMnzmRzneST-oAuA' \
--data-raw '{
  "input": {
    "head-end": "100.100.100.5",
    "end-point": "100.100.100.7",
    "sr-policy-path": {
      "path-optimization-objective": "igp-metric"
    }
  }
}'
```

**Step 3** Note the SID list ID in the API response. You will use this when creating the SID list in the next step. For example:

```
{
  "cisco-crosswork-optimization-engine-sr-policy-operations:output": {
    "segment-list-hops": [
      {
        "step": 0,
        "sid": 23002,
        "ip-address": "100.100.100.7",
        "type": "node-ipv4"
      }
    ],
    "igp-route": [
      {
        "node": "PE-A",
        "interface": "GigabitEthernet0/0/0/0"
      },
      {
        "node": "P-TOPLEFT",
        "interface": "GigabitEthernet0/0/0/2"
```

```
        },
        {
          "node": "P-BOTTOMRIGHT",
          "interface": "GigabitEthernet0/0/0/3"
        }
      ],
      "state": "success",
      "message": ""
    }
  }
```


# Step 2 Create the SID list in the provisioning UI

In this scenario, we will create a SID list for traffic from PE-C to PE-A and another SID list for traffic in the opposite direction.

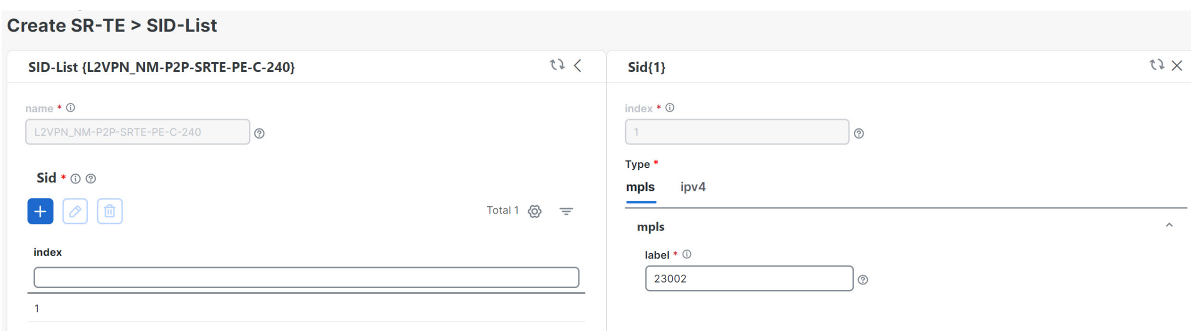
**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **SR-TE** > **SID-List**.

**Step 2** Click + to create a new SID list and give it a unique name. For this example, the SID list name is **L2VPN_NM-P2P-SRTE-PE-C-240**. Click **Continue**.

**Step 3** Under Sid, click + to create a new SID index and give it a numeric value. Click **Continue**.

**Step 4** Under mpls, enter the SID ID that was received in the API response in Step 1.

*Figure 52: SID-List*




**Step 5** Click **X** in the top-right corner to return to the SID list. Your new SID appears in the index table.

**Step 6** Repeat these steps to create additional SID indexes, as required.

**Step 7** Commit your changes.

**Step 8** Check that the new SID list appears in the table.

**Step 9** Create another SID list for the traffic from PE-A to PE-C. For this example, the SID list name is **L2VPN_NM-P2P-SRTE-PE-A-240**.

# Step 3 Create an explicit SR-TE policy for each VPN endpoint by importing a file

In this step, we will provision two explicit SR-TE policies which will reference the SID lists created in Step 1.

The first SR-TE policy specifies PE-C as the headend and provides the IP address of PE-A as the tail end. The second SR-TE policy specifies PE-A as the headend and provides the IP address of PE-C as the tail end.

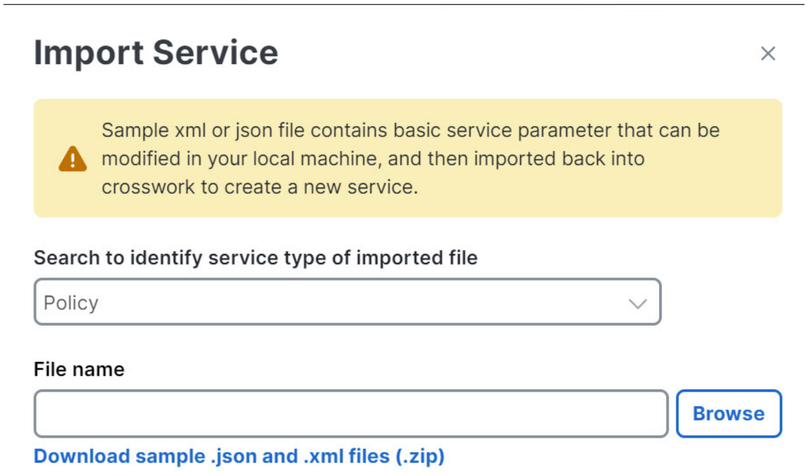
Instead of manually filling in each field in the provisioning UI, we will import an xml file containing all the configurations required to create the SR-TE policy.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **SR-TE** > **Policy**.

**Step 2** Click to import.

**Step 3** Download the sample .json or .xml file, which will serve as a template for the required configuration. In the Import Service dialog, click the **Download sample .json and .xml files (.zip)** link.

***Figure 53: SR-TE Policy Import Service***




**Step 4** Unzip the downloaded file and open sr-Policy.xml in an XML editor.

**Step 5** Edit the xml file as required. Provide a name for the SR-TE policy, and specify the SID list to be associated with this policy. Save the xml file.

*Figure 54: sr-Policy.xml*

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <sr-te xmlns="http://cisco.com/ns/nso/cfp/cisco-tsdn-sr-te">
    <policies xmlns="http://cisco.com/ns/nso/cfp/cisco-tsdn-sr-te-sr-policies">
      <policy>
        <name>SR-Policy-1</name>
        <head-end>
          <name>iosxrv-5</name>
        </head-end>
        <tail-end>7.7.7.7</tail-end>
        <color>100</color>
        <binding-sid>100</binding-sid>
        <path>
          <preference>100</preference>
          <dynamic>
            <metric-type>te</metric-type>
            <metric-margin>
             <relative>40</relative>
            </metric-margin>
            <constraints>
              <sid-limit>10</sid-limit>
            </constraints>
          </dynamic>
        </path>
        <path>
          <preference>200</preference>
          <explicit>
            <sid-list>
              <name>mysidlist</name>
              <weight>10</weight>
            </sid-list>
            <constraints>
              <affinity>
                <rule>
                  <action>include-all</action>
                  <color>GREEN</color>
                  <color>RED</color>
                </rule>
              </affinity>
            </constraints>
          </explicit>
        </path>
      </policy>
      <sid-list>
        <name>mysidlist</name>
        <sid>
          <index>1</index>
          <mpls>
            <label>17001</label>
          </mpls>
```

**Step 6** In the Import Service dialog, select **Policy** as the type of file to import, browse to the edited xml file, and click . If there are any errors in the file, you will receive a notification. If there are no errors, the file will be imported, and the policy and the devices will be configured accordingly.

**Step 7** Check whether the new SR-TE policy appears in the Policy table and whether its Provisioning State is **Success**.

**Step 8** Click  in the Actions column and choose **Config View** to see the Yang model-based service intent data that details the SR-TE policy you created. You can also check the devices themselves to ensure they were provisioned correctly.

# Step 4 Create and provision the L2VPN service

In this step, we will create and provision a P2P VPN service with PE-A and PE-C as the endpoints. The VPN service will reference the SR-TE policies we created in the previous step to ensure that the traffic traversing the VPN will follow the path defined in the SID lists.

As we did with the SR-TE policy, we will create the VPN service by importing an xml file containing all the required configurations. Once we have provisioned the VPN service, we will edit it in the provisioning UI to associate the SR-TE policies.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L2vpn** > **L2vpn-Service**.

**Step 2** Click to import.

**Step 3** If you did not download the sample .json or .xml files in Step 3, do so now.

**Step 4** Open l2vpn-service.xml in an XML editor.

**Step 5** Edit the xml file as required. Provide a name for the L2VPN, configure each endpoint, and define the VPN parameters.

This is the configuration for PE-A in our example:

***Figure 55: l2vpn-service.xml***

```
    <vpn-node-id>xrv9k-22</vpn-node-id>
    <signaling-option>
      <ldp-or-l2tp>
        <pw-peer-list>
          <peer-addr>192.168.0.22</peer-addr>
          <vc-id>100</vc-id>
          <mpls-label xmlns="http://cisco.com/ns/nso/fp/examples/cisco-l2vpn-ntw">100</mpls-label>
        </pw-peer-list>
      </ldp-or-l2tp>
    </signaling-option>
    <vpn-network-accesses>
      <vpn-network-access>
        <id>300</id>
        <interface-id>GigabitEthernet0/0/0/1</interface-id>
        <connection>
          <encapsulation>
            <encap-type xmlns:vpn-common="urn:ietf:params:xml:ns:yang:ietf-vpn-common">vpn-common:dot1q</encap-type>
            <dot1q>
              <cvlan-id>100</cvlan-id>
            </dot1q>
          </encapsulation>
        </connection>
      </vpn-network-access>
    </vpn-network-accesses>
    <te-service-mapping xmlns="http://cisco.com/ns/nso/fp/examples/cisco-l2vpn-ntw">
      <te-mapping>
        <sr-policy>
          <policy-type>policy</policy-type>
          <policy>SR-300</policy>
        </sr-policy>
      </te-mapping>
    </te-service-mapping>
  </vpn-node>
  <vpn-node>
    <vpn-node-id>xrv9k-23</vpn-node-id>
```

**Step 6** Save the xml file.

**Step 7** In the Import Service dialog, select **l2vpn service** as the type of file to import, browse to the edited xml file, and click . If there are any errors in the file, you will receive a notification. If there are no errors, the file will be imported, and the policy and the devices will be configured accordingly.

**Step 8** Check that the new L2VPN service appears in the L2VPN Service table and its Provisioning State is **Success**.

**Step 9** Click in the Actions column and choose **Config View** to see the Yang model-based service intent data that details the VPN service you created. You can also check the devices themselves to ensure that they were provisioned correctly.

# Step 5 Attach the SR-TE policies to the L2VPN service

At this stage, the provisioned L2VPN service you created does not have associated SR-TE policies that define the transport path. In this step, we will edit the L2VPN service in the provisioning GUI, attach the relevant SR-TE policies to each endpoint, and re-provision it.

**Step 1** Locate the L2VPN in the VPN Service table.

**Step 2** Click ⋯ in the Actions column and choose **Edit**.

**Step 3** Under vpn-nodes, select **PE-A** and click the **Edit** button above the table.

**Step 4** In the pane that opens on the right, open the **te-service-mapping** > **te-mapping** section.

**Step 5** In the sr-policy tab, you can find the list of configured SR policies in the **policy-type** drop-down list. Choose the policy created for PE-A: **L2VPN_NM-P2P-SRTE-PE-A-240** from the list or enter the name of the SR-TE policy in the **policy** field.

**Step 6** Click **X** in the top-right corner to close the PE-A pane.

**Step 7** Repeat the above steps for PE-C and attach the SR-TE policy: **L2VPN_NM-P2P-SRTE-PE-C-240**.

**Step 8** Click **Commit Changes**.

# Step 6 Enable Service Health monitoring

After creating and provisioning the required L2VPN services, you can begin monitoring their health.

### Before you begin

- Ensure that Crosswork Service Health is installed. For details, see the Install Crosswork Applications chapter in Crosswork Network Controller 7.0 Installation Guide.
- Ensure that the required L2VPN services are created and provisioned.

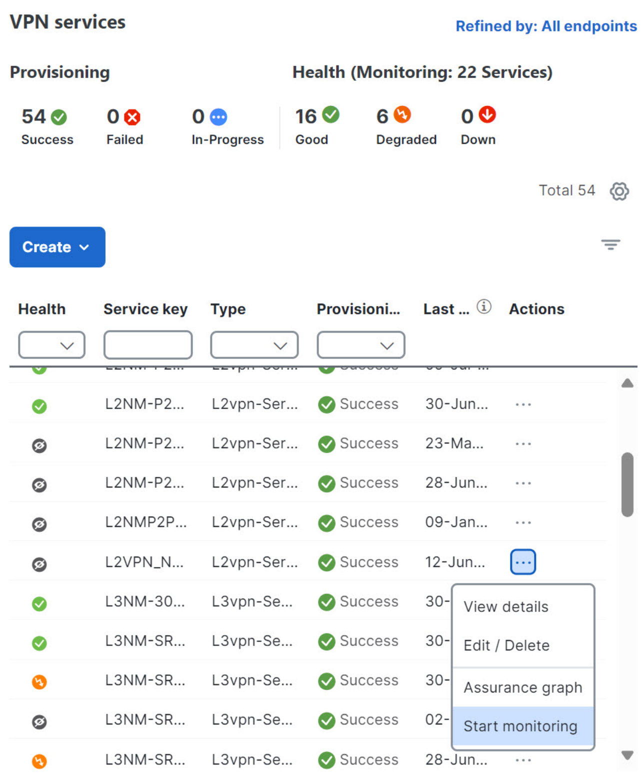
To enable service health monitoring, do the following:

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The map opens on the left side of the page, and the table opens on the right side.

**Step 2** In the Actions column, click ⋯ for the service you want to start monitoring for health.

**Step 3** Click **Start Monitoring**.

*Figure 56: Start Monitoring*

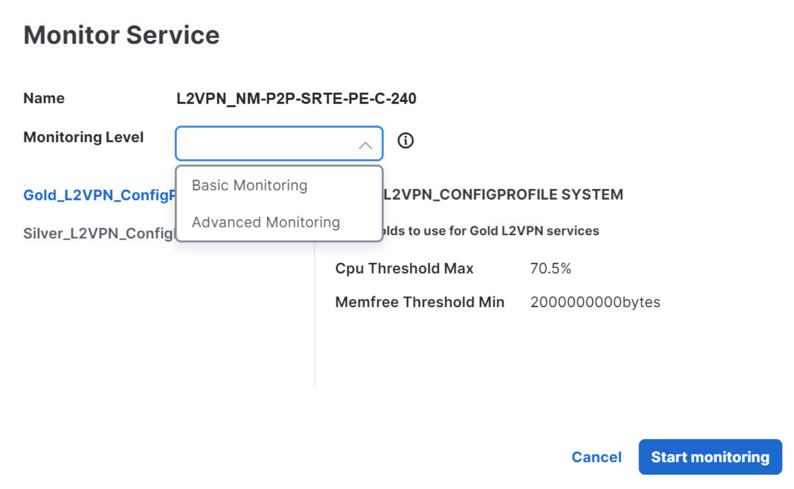



**Note** The Health column color coding indicates the health of the service:

- Blue = Initiated
- Green = Good
- Orange = Degraded
- Red = Down
- Gray = Not Monitoring

**Step 4** In the Monitor Service dialog box, select the Monitoring Level. For help choosing the appropriate monitoring level for your needs, see Crosswork Network Controller 7.0 Service Health Guide reference chapter, Basic Monitoring and Advanced Monitoring Rules.

*Figure 57: Basic and Advanced Monitoring Rules*




Once you have started monitoring the health of this service, select the Actions column and click ⋯ to view additional Service Health options. You will see **Stop Monitoring**, **Pause Monitoring**, **Edit Monitoring Settings**, and **Assurance Graph** options.

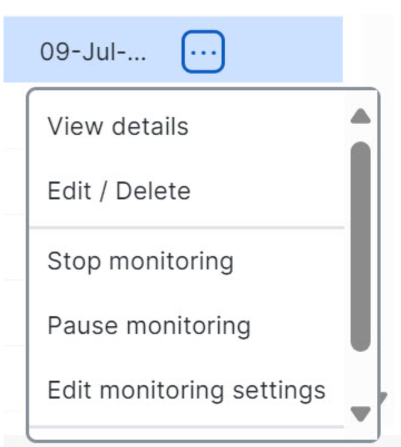
**Note** If you select **Edit Monitoring Settings**, you may update the Monitoring Level setting, from Basic Monitoring to Advanced Monitoring, or from Advanced Monitoring to Basic Monitoring at any time.

**Note** If you later decide to **Stop Monitoring** a service that has already been started, you have the option to retain the historical service data for that stopped service. See Crosswork Network Controller 7.0 Service Health Guide, Stop Service Health Monitoring section for more details.

**Step 5** Click **Start Monitoring**.

**Note** Once you have started monitoring the health of this service, select the Actions column and click ⋯ to view additional Service Health options. You will see **Stop Monitoring**, **Pause Monitoring**, **Edit Monitoring Settings**, and **Assurance Graph** options.

*Figure 58: Service Health Actions Options*



**Step 6** Repeat these steps for each service for which you wish to start health monitoring.

**Step 7** Click **X** in the top-right corner when you are done.

# Step 7 Visualize the L2VPN on the map

In this step, we will examine the representation of the L2VPN on the map and see the paths the traffic will take from PE-A to PE-C and vice versa based on the explicit SR-TE policies we created.

**Step 1** In the L2VPN Service table, in the Actions column for the new VPN, click ⋯ and choose **ViewDetails** from the menu. The map opens, and the service details are shown to the right of the map.

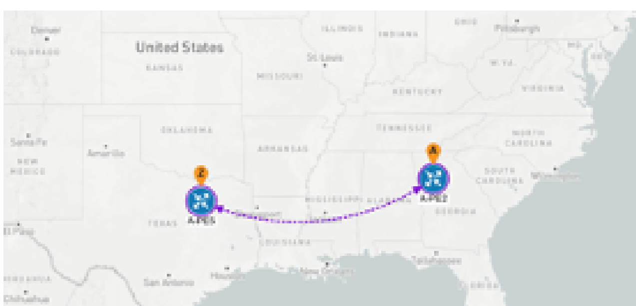
or

From the main menu, choose **Services & Traffic Engineering** > **VPN Services**.

The map opens, and a table of VPN services is displayed to the right of the map.

a) Click on the VPN in the Services table. If there are many services in the table, you can filter by name, type, or provisioning state to help locate the VPN.
b) In the map, you will see the VPN as an overlay on the topology. It shows a representation of the endpoints and a solid line that indicates that it is a virtual path.
c) Select the **Show Participating Only** check box if you do not want to see the devices that are not involved in the selected VPN.

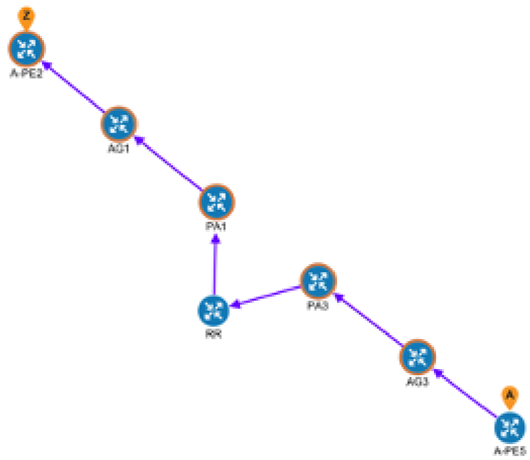
***Figure 59: Visualize the L2VPN on the map***




**Step 2** Under the Actions column, click ⋯ and choose **View Details** to drill down to a detailed view of the VPN service, including the device configurations, the computed transport paths, and the health status for transport paths.

**Step 3** In the Transport tab, select one or more SR-TE policies to see the path from endpoint to endpoint on the map. The image below shows the path from PE-C to PE-A. The **Show IGP Path** check box in the top left corner of the map is selected so the physical path is shown. The dashed line indicates that this link is being used to transport multiple services.

*Figure 60: SR-TE policies physical path*




# Step 8 Inspect a degraded service using Service Health and last 24Hr metrics to identify issues

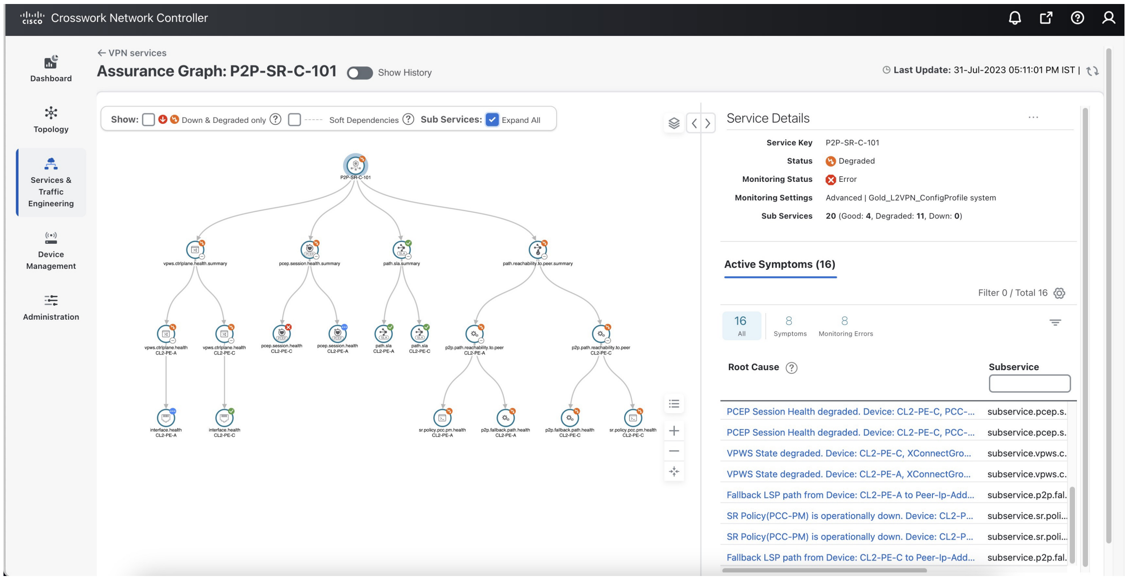
In this step, you can utilize the Last 24Hr Metrics to identify the issues with the degraded services within a specific time range. By narrowing down the issues within a particular time frame, you can focus on the details that may have caused the degraded (or down) service. This will help in troubleshooting the service or the node to address specific symptoms in detail.

**Step 1** Return to the VPN Services list.

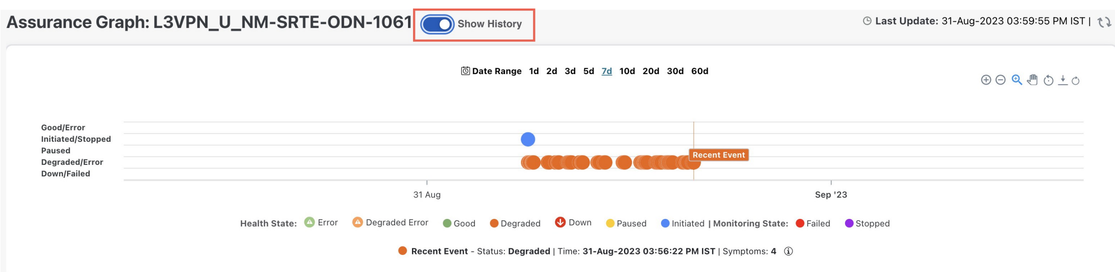
**Step 2** In the Actions column, click ... for the degraded service and click **Assurance Graph**. The topology map of services and subservices appear with the Service Details panel showing Service Key, Status, Monitoring Status, Monitoring Settings, Sub Services, and Active Symptoms details.

**Note** This may take up to 5-10 minutes to update after a service has been enabled for monitoring.

*Figure 61: Assurance graph*



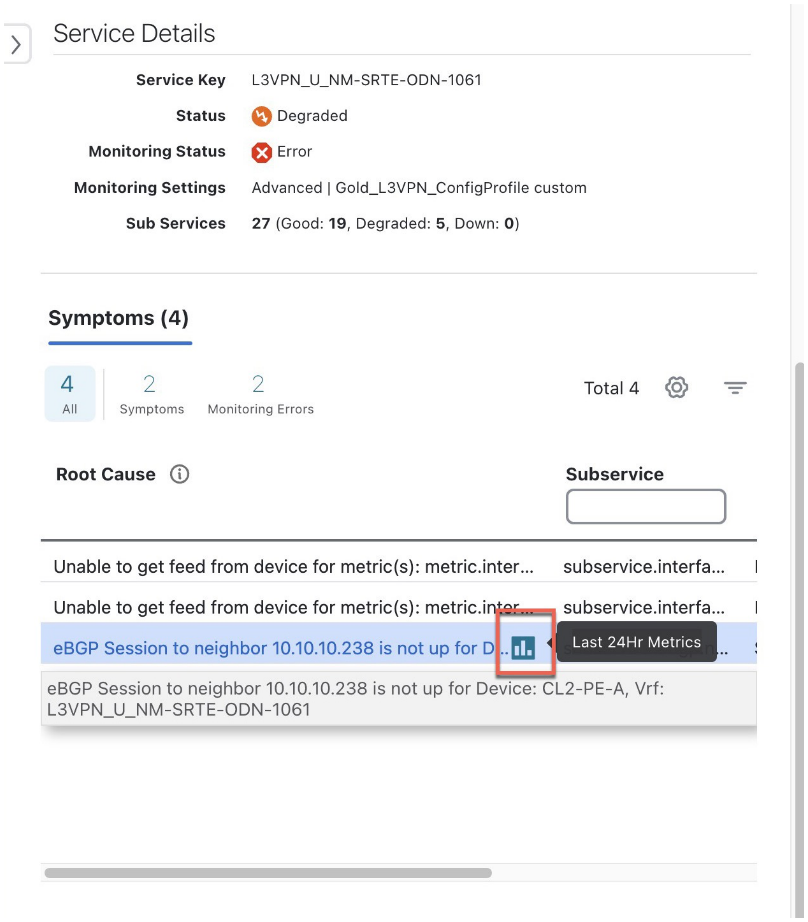

**Step 3** At the top of the page, click the **Show History** mode toggle. The historical Date Range graph appears. This graph shows different ranges of historical health service monitoring details from one day (1d) to sixty days (60d). When you hover over an event on the Date Range graph, a tooltip with information about that event appears (such as the date and time of the event and several symptoms).




**Step 4** You can review the Root Cause information by clicking a particular event in the graph. The Service Details panel reloads, showing the active symptoms and the root causes associated with the event. You can resize columns using your mouse, or you can select the gear icon to deselect or select columns you want to appear.

**Note** Once you enable **Show History** mode, Root Cause information in the Active Symptoms table will show the blue Last 24Hr Metrics icon. However, data from the device will be initially delayed and may take some time before the **Last 24Hr Metrics** begin to populate with data. Until then, the value of zero is reported.

*Figure 62: Last 24Hr metrics*



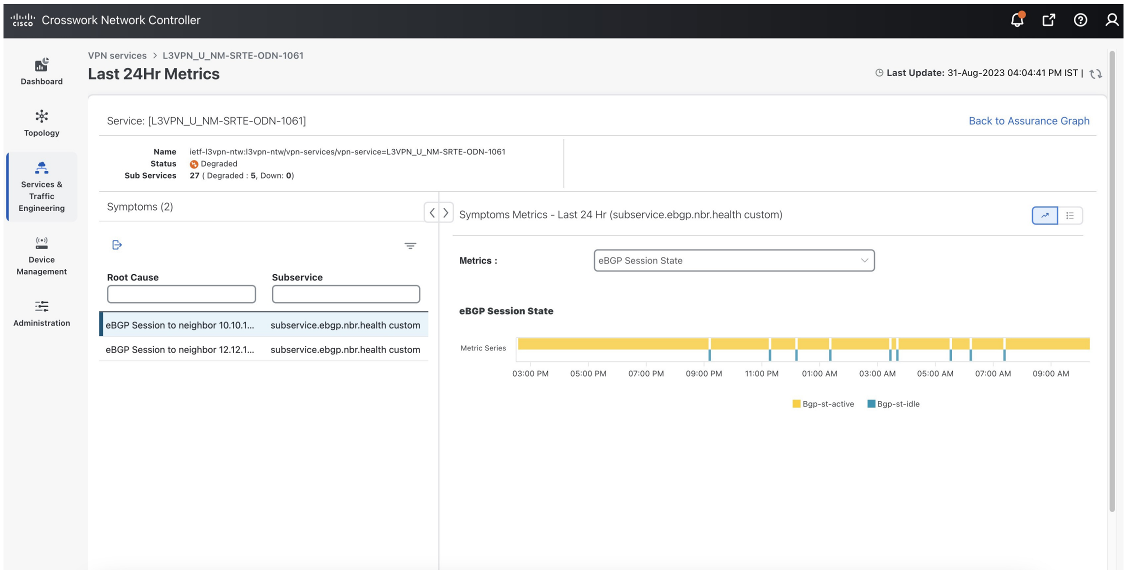

**Step 5** To further isolate the possible issues and to utilize the **Last 24Hr Metrics**, perform the following steps:

a) In the Date Range graph, use your mouse to select the range of historical health service monitoring details from one day (1d) to sixty days (60d).

**Note** At the top-right of the Date Range graph, select the appropriate icons to zoom in or out, scroll through the date ranges horizontally, or refresh the graph to return to the most recent event, for example. You can also use your mouse to draw a rectangle over events to further zoom in on the degraded devices. Consecutive events may appear as a line of white space.

b) Click on a degraded event in the graph. The Service Details panel reloads, showing any active symptoms and the root causes to be inspected. Expand the table and information as necessary for further details.

*Figure 63: Active symptoms and the root causes*




**Step 6** Check the **Down & Degraded Only** check box at the top-left corner of the map to show only the Subservices that are degraded, along with other dependent but healthy subservices. Inspect the Service Details panel showing the active symptoms and their root cause. Uncheck the **Down & Degraded Only** check box and check the **Soft Dependencies** check box in the top-left corner of the map. Soft Dependencies implies that a child subservice's health has a weak correlation to its parent's health. As a result, the degraded health of the child will not result in the parent's health degradation.

Use the **+** or **–** symbols in the bottom-right corner of the map to zoom in or out on services mapped. Select the **?** to view the Link Color Legend that explains all the icons, symbols, badges, and colors and their definitions.

**Step 7** Select the degraded subservice in the map to show the subservice details.

**Step 8** Click the **Symptoms** tab to show any root causes for the service health details, and then click the **Impacted Services** tab to view the impacted services.

**Step 9** Click **X** in the top-right corner to return to the VPN Services list. In the Actions column, click  for the degraded service and click **Assurance Graph** to show the Service Details panel.

**Step 10** Again, select the **Show History** toggle in the top-right corner of the Service Details panel before selecting the blue metrics icon in one of the Root Cause rows. The Symptoms Metrics – Last 24 Hr bar chart appears. This chart provides details of the metric patterns for different session states (such as active, idle, failed) for individual root cause symptoms with Status, Session, Start Time, and Duration information to assist in troubleshooting prevailing issues. Use your mouse to hover over the chart to view the different details.

**Continue to troubleshoot a service health issue using Parameterized Jobs**

To further troubleshoot a service health issue (such as a degraded device due to improper data fetching), continue with the following steps to determine whether the problem is related to a collection job.

**Step 11** From the main menu, choose **Administration** > **Collection Jobs**.

The Collection Jobs page appears.

**Step 12** Click the **Parameterized Jobs** tab.

**Step 13** Review the Parameterized Jobs list to identify devices with service health degradation issues. By reviewing Parameterized Jobs, you can identify and focus on gNMI, SNMP, and CLI-based jobs by their Context ID (protocol) for further troubleshooting purposes.

**Step 14** In the Job Details panel, select the collection job you want to export and download the status of collection jobs for further examination. The information provided is collected in a .csv file when the export is initiated.

> **Note** When exporting the collection status, you must fill in the information each time an export is executed. In addition, make sure to review the **Steps to Decrypt Exported File** content available on the Export Collection Status dialog box to ensure you can access and view the exported information.

**Step 15** Click **Export**.

**Step 16** To check the status of the exported collection job data, click **View Export Status** at the top right of the Job Details panel. The Export Status Jobs panel appears, providing the status of the export request.

**Step 17** Review the exported .csv file for collection job details and the possible cause of the degraded device.

## Summary and conclusion

In this scenario, we observed how simple it is to create explicit SR-TE policies and attach them to an L2VPN service to mandate a static path for mission-critical traffic. We saw how editing and importing a pre-defined template into the system enables quick and easy provisioning of services and SR-TE policies. We were then able to visualize the actual traffic paths on the map. Lastly, we used Service Health to monitor the health of the new service using the Assurance Graph, Last 24hr Metrics, and SubExpressions metrics to view when service may have been up, degraded, or down, and what the root causes were identified.

# Scenario: Provision an L2VPN service over an RSVP-TE tunnel with reserved bandwidth

For the continuous stream transmission required for rich data media types, such as video and audio, bandwidth reservation is often required to provide a higher quality of service. Crosswork Network Controller supports creating and managing RSVP-TE tunnels to reserve guaranteed bandwidth for an individual flow. RSVP is a per-flow protocol that requests a bandwidth reservation from every node in the flow path. The endpoints, or other network devices on behalf of the endpoints, send unicast signaling messages to establish the reservation before the flow is allowed. If the total bandwidth reservation exceeds the available bandwidth for a particular LSP segment, the LSP is rerouted through another LSR. If no segments can support the bandwidth reservation, the LSP setup fails, and the RSVP session is not established.

In this scenario, we will:

- Create RSVP-TE tunnels with reserved bandwidth.
- Enable Bandwidth on Demand functionality.
- Provision a VPN service from PE-A to PE-B and attach the RSVP-TE tunnels as underlay configuration.
- Visualize the traffic path when link utilization is below the bandwidth threshold. This path would change if the bandwidth utilization on the link crossed the specified threshold.

### Assumptions and prerequisites

The following are the assumptions and prerequisites for provisioning an L2VPN service over an RSVP-TE Tunnel with reserved bandwidth.

- For transport mapping to L2VPN service, devices must be configured with the **l2vpn all** command.
- For Service Health enablement, Service Health must be installed. For instructions, see Crosswork Network Controller Installation Guide chapter, Install Crosswork Applications.
- For steps to enable Service Health during this scenario, see Scenario 3, Step 6 Enable Service Health monitoring. For additional details on Service Health, see Scenario: Implement and maintain SLA for an L3VPN service for SR-MPLS (using ODN) and Scenario: Mandate a static path for an EVPN-VPWS service using an explicit MPLS SR-TE policy.
- (Optional) Service Health provides **Internal Storage** of monitoring data up to a maximum limit of 50 GB. This data is stored on your system. If you exceed the internal storage limit, the least recently used historical data will be lost. If you extend Service Health storage capacity, you can configure **External Storage** in the cloud using an Amazon Web Services (AWS) cloud account. By leveraging External Storage, all existing internal storage data will be automatically moved to the external cloud storage (see Crosswork Network Controller 7.0 Service Health Guide section, Configure Additional External Storage for more details), and your internal storage will act locally as cache storage. Configuring external storage for Service Health ensures you will not lose historical data for services that continue to monitor a service's health and will retain service health data for any service you choose to stop monitoring when you select the option to retain historical monitoring service for the data. For more information on Internal and External Storage and how to retain historical monitoring service data when stopped, see Crosswork Network Controller 7.0 Service Health Guide section, Stop Service Health Monitoring.
- (Optional) For initializing a Heuristic Package to monitor the health of the services, see Crosswork Network Controller 7.0 Service Health Guide chapter, Customize Heuristic Packages for detailed steps to be performed before starting monitoring.

# Step 1 Create an RSVP-TE tunnel for both directions of the L2VPN

In this step, we will create an RSVP-TE tunnel from PE-A to PE-B and from PE-B to PE-A, and we'll reserve bandwidth of 1200 on the link.

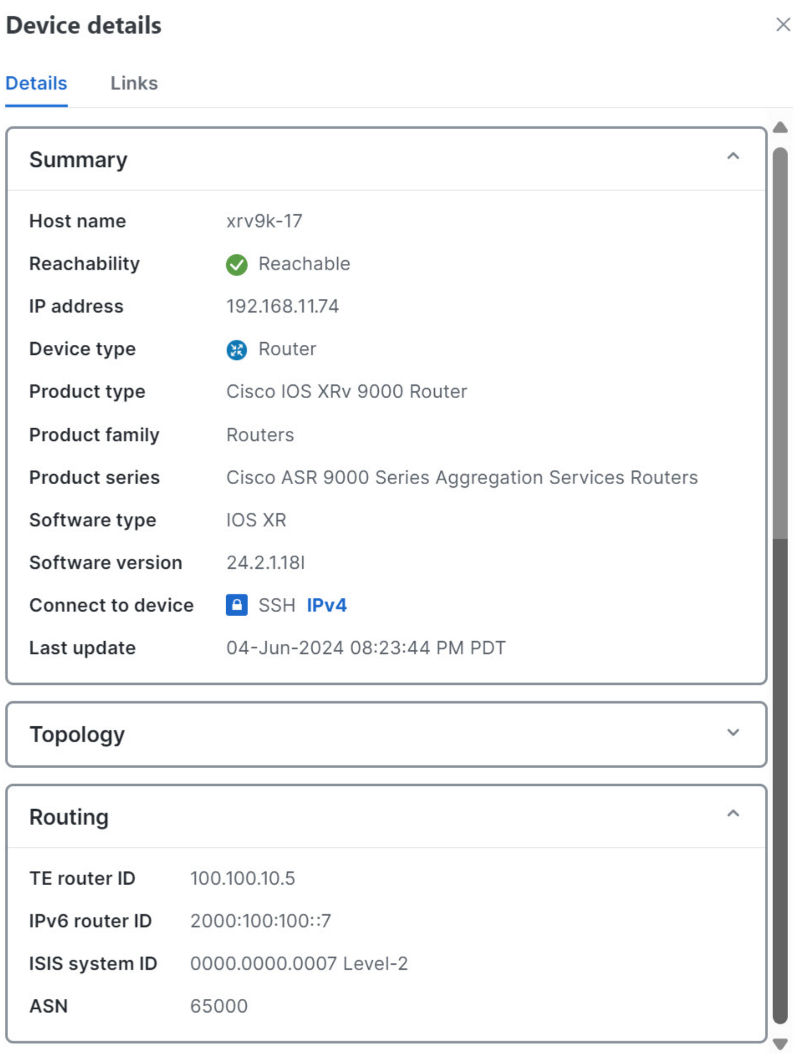
**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **RSVP-TE** > **Tunnel**.

**Step 2** Click + to create a new RSVP-TE tunnel and give it a unique name. Click **Continue**.

**Step 3** In the **identifier** field, enter a numeric identifier for the tunnel. You will use this identifier to associate this RSVP-TE tunnel with the L2VPN service. For this example, the identifier is **2220**.

**Step 4** In the source and destination fields, enter the loopback0 IP address of the source (PE-A) and the destination (PE-B) devices. This is the TE router ID. To find the TE router ID, go to Topology and click on a device in the map or the list of devices. The Device Details pane opens, and the TE router ID is shown in the Routing section.

*Figure 64: TE router ID in device details*




**Step 5** Define the endpoints:

a) Under head-end, select the headend device from the dropdown list.
b) Under tail-end, select the tailend device from the dropdown list.

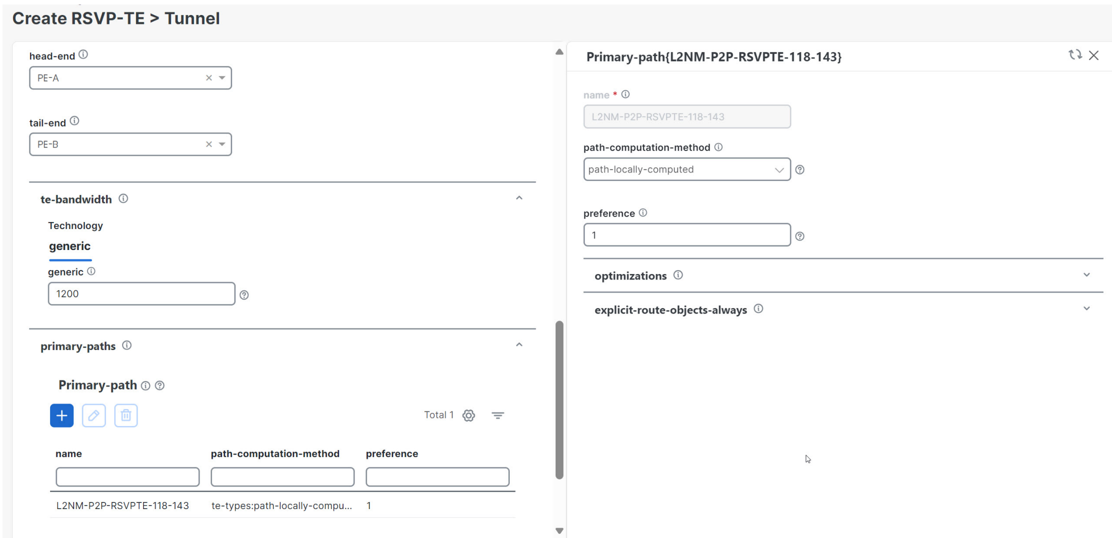
**Step 6** Reserve bandwidth on the link. Under te-bandwidth > generic, enter the bandwidth threshold for the link.

**Step 7** Define the path of the RSVP-TE tunnel.

You can define an explicit path or have the path locally computed by the participating devices. Alternatively, you can have the SR-PCE compute a path dynamically. For this scenario we will have the path locally computed.

a) Under Primary-paths, click + to create a new path.
b) Provide a path name in the pane that appears on the right.
c) Select the path computation method – **path-locally-computed**.
d) Specify a numeric preference for the path. The lower the number, the higher the preference.
e) Define the optimization metric, in this case, **igp**.

*Figure 65: Create RSVP-TE tunnel page*

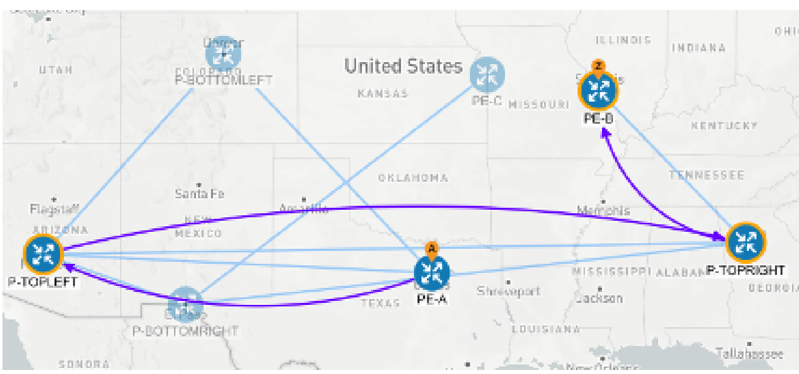



**Step 8** Click **Commit Changes**.

**Step 9** Verify that the RSVP-TE tunnel appears in the list of tunnels and that its Provisioning State is **Success**.

**Step 10** Click on the tunnel name to visualize the tunnel on the map and to see the tunnel details.

*Figure 66: RSVP-TE tunnel visualization*




# Step 2 Create the L2VPN service and attach the RSVP tunnel to the service

In this step, we will create a P2P L2VPN service using the provisioning GUI. If you want to create the service by importing a template, refer to the third scenario, *Mandate a static path for an EVPN-VPWS service using an explicit SR-TE policy.*

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **L2VPN** > **L2vpn Service**.

**Step 2** Click + to create a new service and give it a unique name. Click **Continue**.

**Step 3** Choose the vpn-type field.

**Step 4** Define each VPN endpoint individually – PE-A and PE-B.

a) Under vpn-nodes, click +.

b) Select the relevant device from the vpn-node-id and ned-id dropdown lists and click **Continue**.

**Step 5** Define the LDP signaling options by creating one or more pseudowires. In this case, specify the TE router ID of the peer device (PE-B), and provide a unique numeric label to identify the pseudowire.

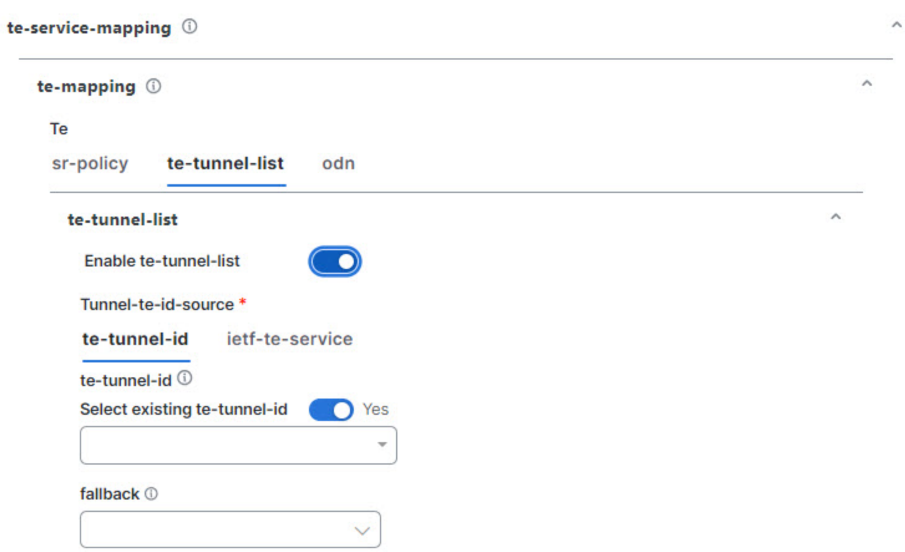
**Step 6** Attach the RSVP tunnel to the service:

a) Under te-service-mapping > te-mapping, click the te-tunnel-list tab.

b) Click the **te-tunnel-id** tab.

c) If you have an RSVP-TE tunnel on the device that was configured externally to Crosswork Network Controller, you can provide that tunnel ID under this tab. Use the toggle button next to the **Select existing te-tunnel-id** field to choose the RSVP-TE tunnel from a list of configured tunnels or manually enter the tunnel ID.

***Figure 67: TE Service Mapping to RSVP Tunnel***




d) If you want to specify the IETF-TE service name from which te-tunnel-id will be extracted, click the **ietf-te-service** tab and specify the service name. The tunnel ID will be extracted from the tunnel configuration.

**Step 7** Define the VPN network access. In this case, we are using dot1q encapsulation and we have specified the physical interface (GigabitEthernet0/0/0/2) and the VLAN ID (2220).

**Step 8** Follow the above steps for PE-B as well.

**Step 9** Click **Commit Changes**. Verify that the L2VPN appears in the list of VPN services and that its Provisioning state is **Success**.

# Step 3 Visualize the L2VPN service on the map

In this step, we look at the representation of L2VPN on the map and see the paths the traffic will take from PE-A to PE-B and vice versa, based on the RSVP-TE tunnels we created.

**Step 1** In the L2VPN Service table, click on the service name. The map opens, and the service details are shown to the right of the map.

or

a) From the main menu, choose **Services & Traffic Engineering > VPN Services**.

The map opens, and a table of VPN services is displayed to the right of the map.

b) Click on the VPN in the Services table. When there are many services in the table, you can filter by name, type, or provisioning state to help locate the VPN.

In the map, the VPN is an overlay on the topology. It shows a representation of the three endpoints and a dashed line indicating that it is a virtual path. Use the buttons at the top right of the map to toggle between the logical and geographical maps.

**Step 2** To see the hops in the route between PCC7_56 and PCC5_81, click the Transport tab and select one or more of the underlying TE tunnels to see the path from endpoint to endpoint on the map.

**Step 3** As the RSVP-TE tunnels are configured with a reserved bandwidth, if the bandwidth utilization across the link exceeds the specified bandwidth, the path would be rerouted.

## Summary and conclusion

This scenario illustrated how to create RSVP-TE tunnels with reserved bandwidth and attach them to an L2VPN service to meet the high quality of service requirements for continuous streaming of rich data media. We observed the path on the map. This path would be recomputed if the bandwidth utilization on the link crossed the bandwidth reservation threshold.

# Scenario: Provision a soft bandwidth guarantee with optimization constraints

Service providers must be able to provide fast connections with the lowest latency possible to meet the needs of customers' peak traffic utilization times and to dynamically optimize services based on the customers' changing priorities throughout the day. For this purpose, the operator might need to reserve bandwidth on specific links to ensure a dedicated path that can handle a set amount of traffic with a specific optimization intent. The Bandwidth on Demand (BWoD) feature within Crosswork Network Controller enables this functionality. Paths with the requested bandwidth are computed when available. If a path that guarantees the requested bandwidth cannot be found, an attempt will be made to find a *best effort* path.

In this scenario, we will use BWoD to calculate the lowest TE metric path with a specified amount of available bandwidth between two endpoints.

This scenario uses the following topology as a base:

*Figure 68: Base topology*


The goal is to create a path from F2.cisco.com to F7.cisco.com that can accommodate 250 Mbps of traffic while keeping the utilization at 80%. BWoD will initially try to find a single path to accommodate the requested bandwidth without exceeding the utilization threshold. If a single path cannot be found, BWoD may recommend splitting the path.

In this scenario, we will:

- Orchestrate a new SR-TE policy with bandwidth and TE constraints.
- Configure and enable BWoD.
- Verify the state of the SR-TE policy and view the path on the map.

# Step 1 Create a BWoD SR-TE policy with the requested bandwidth and optimization intent

In this step, we will create a BWoD SR-TE policy with requested bandwidth and optimization intent.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)** > **SR-TE** > **Policy**.

**Step 2** Click + to create a new SR-TE policy and give it a unique name. Click **Continue**.

**Step 3** Define the endpoints:

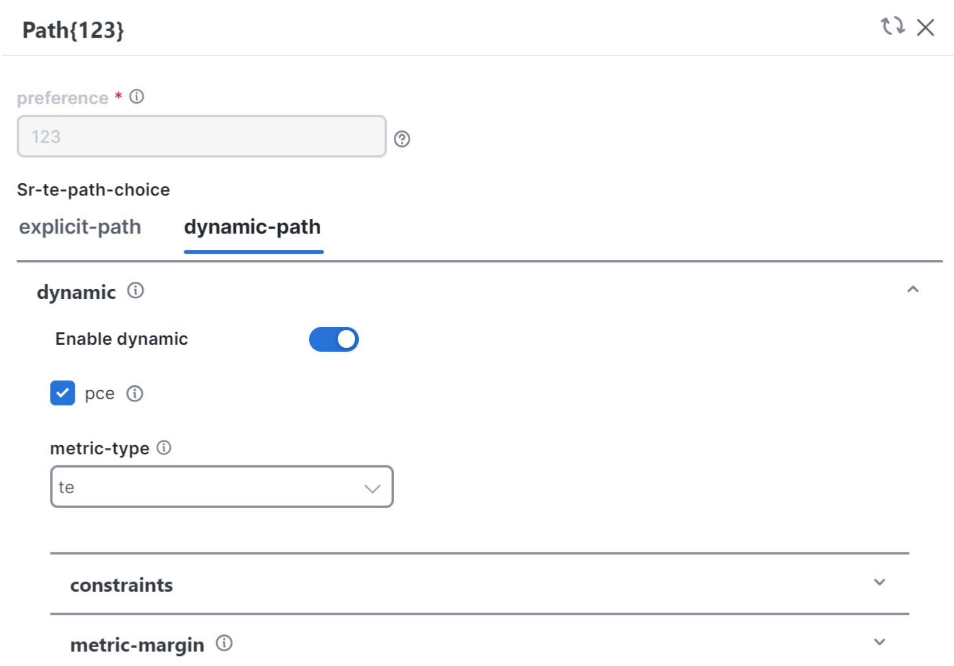
a) Under head-end, click + and select the headend device from the dropdown list and click **Continue**. Click **X** to close the Headend pane.
b) Enter the IP address of the tail-end device.
c) Enter a color to identify the traffic.

**Step 4** Define the parameters on which the path will be computed:

a) Under path, click +.
b) Enter a path preference and click **Continue**.

c) In the dynamic-path tab, select **te** in the metric-type dropdown list as the optimization objective.
d) Select the **pce** check box to have the SR-PCE compute the paths for this policy.

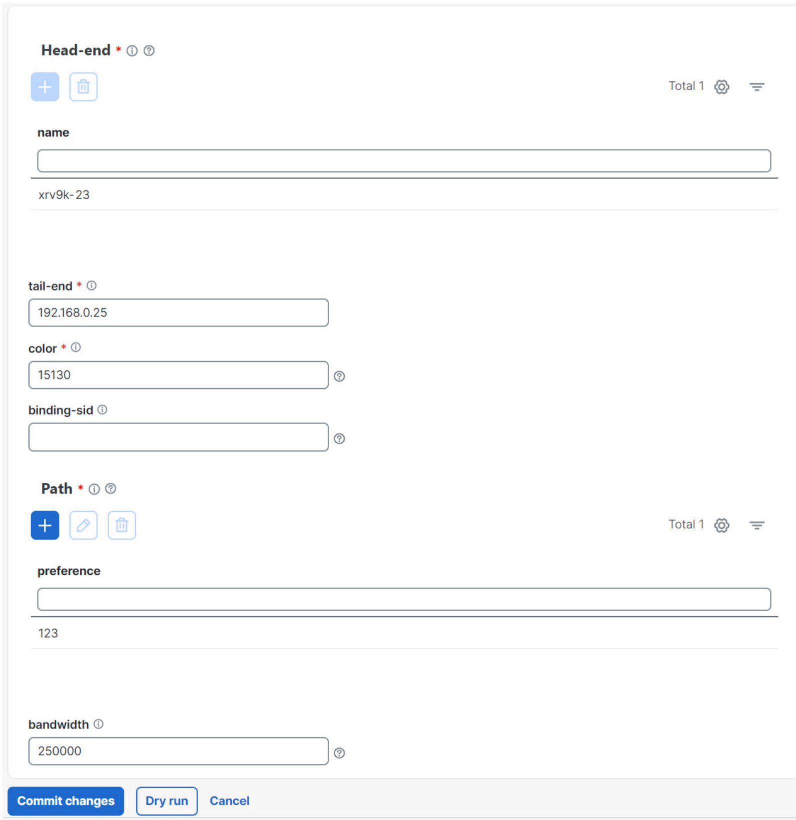
***Figure 69: SR-TE policy path parameters***




e) Click **X** to close the path pane.

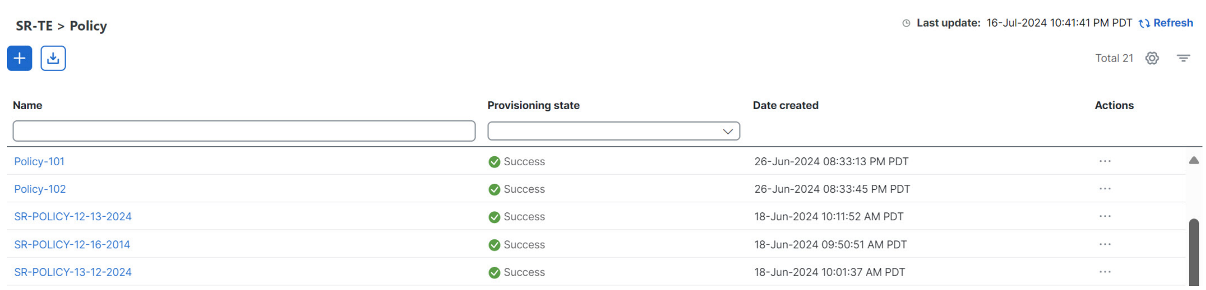
**Step 5** In the **Bandwidth** field enter the requested bandwidth in Kbps. In this case, we are requesting **250** Mbps or 250,000 Kbps.

*Figure 70: SR-TE policy parameters*




**Step 6** Click **Commit changes**. The new policy is created and appears in the list of SR-TE policies. The provisioning state should be **Success**.

*Figure 71: SR-TE policy*




**Step 7** Verify the new policy by viewing its details and its representation on the map:

a) Click ... in the Actions column and choose **View**.

b) The map opens with the SR-TE policy details displayed to the right of the map.

**Note** The policy's operational state is down because the SR-PCE alone cannot address bandwidth computations before the BWoD functionality within Crosswork Network Controller is enabled.

# Step 2 Enable and configure BWoD

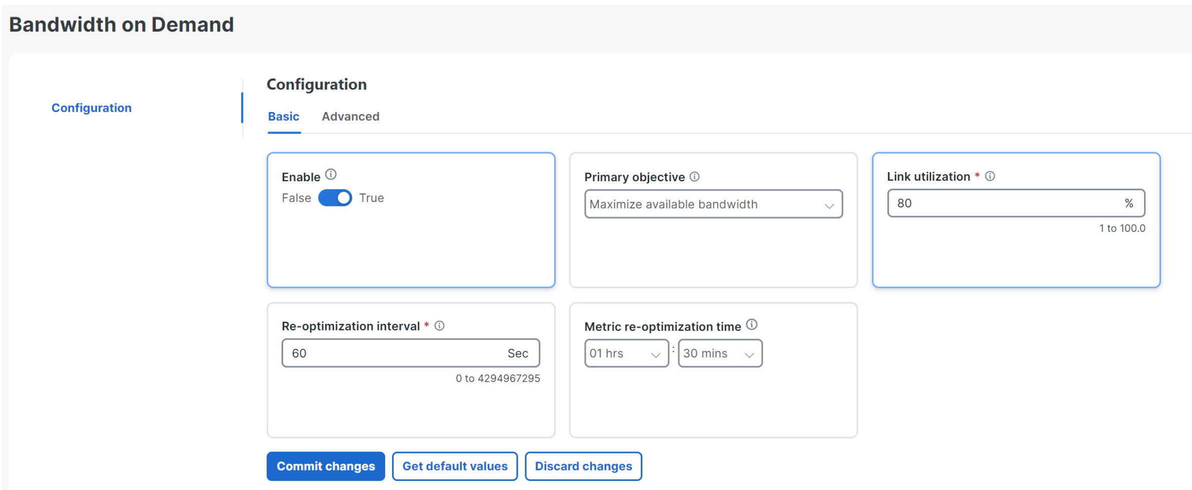
In this step, we will enable and configure BWoD.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Bandwidth on Demand**.

**Step 2** Toggle the Enable switch to True, and enter 80 to set the utilization threshold percentage. To find descriptions of other options, hover the mouse over.

**Step 3** Click **Commit changes**.

*Figure 72: Bandwidth on demand*




# Step 3 Verify that the policy's operational state is now up and view the path on the map

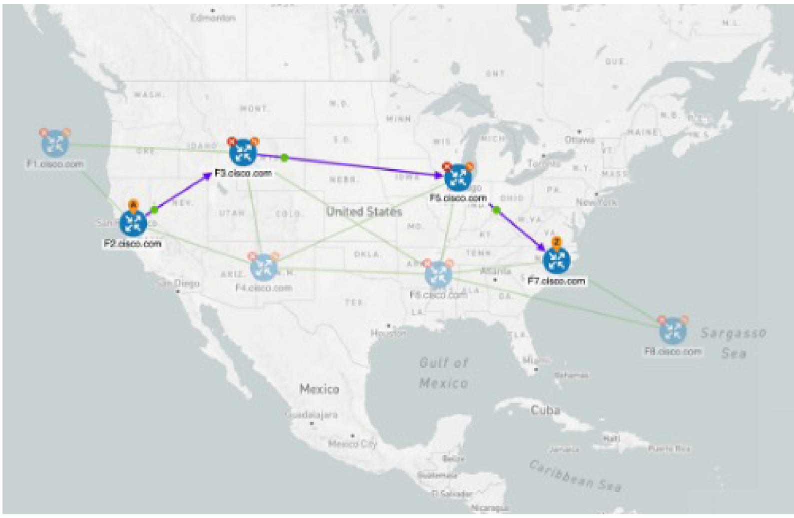
In this step, we verify the policy's operational state is Up and inspect the path on the map for verification.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)**.

**Step 2** In the Policy table, locate and select the path computed for the endpoints.

**Step 3** The path is shown as an overlay on the map. Select the **IGP Path** check box to see the physical path between the endpoints.

*Figure 73: Computed path on the map*




# Summary and conclusion

Operators can set and maintain bandwidth requirements based on optimization intent using the BWoD functionality provided in Crosswork Network Controller. This scenario illustrated how to provision an SR-TE policy with a specific bandwidth request. We saw how to enable BWoD functionality to reroute traffic automatically to maintain bandwidth requirements. This automation alleviates manually tracking and configuring paths to accommodate bandwidth requirements set by SLAs.

CHAPTER 4

# Bandwidth and Network Optimization

This section explains the following topics:



# Overview

### Objective

Network operators need a toolset to help automate bandwidth optimization, steer traffic with little operator intervention, and ensure that critical links always have sufficient bandwidth to avoid congestion.

### Challenge

For service providers, managing bandwidth problems used to be a reactive and manual process. Pressure to solve it is huge. Network congestion leads to poor end-customer experiences. Congested links, high latency, and other network impairments lead to a poor perception of the services carried across your network or result in an inability to meet the service level agreements (SLAs) you have with your customers. In the worst-case scenario, your network issues lead to SLA or contract violations and the loss of your brand equity.

### Solution

Using LCM and Circuit Style policies, service providers can now specify business-critical links with the intention of reserving bandwidth for these links. Identifying critical links and the operator's intention enables automatic optimization of the network in real-time.

Crosswork Network Controller offers both:

- Local Congestion Mitigation (LCM) is a tactical solution for bandwidth management and congestion mitigation. It is best applied when attempting to solve congestion issues directly, on the devices themselves, without a full-scale traffic matrix or advanced planning.
- Circuit-Style Segment Routing (CS-SR) is a strategic traffic engineering solution that permits you to reserve bandwidth in advance for critical links, avoiding congestion issues entirely for these high-priority links.

### Local Congestion Mitigation (LCM)

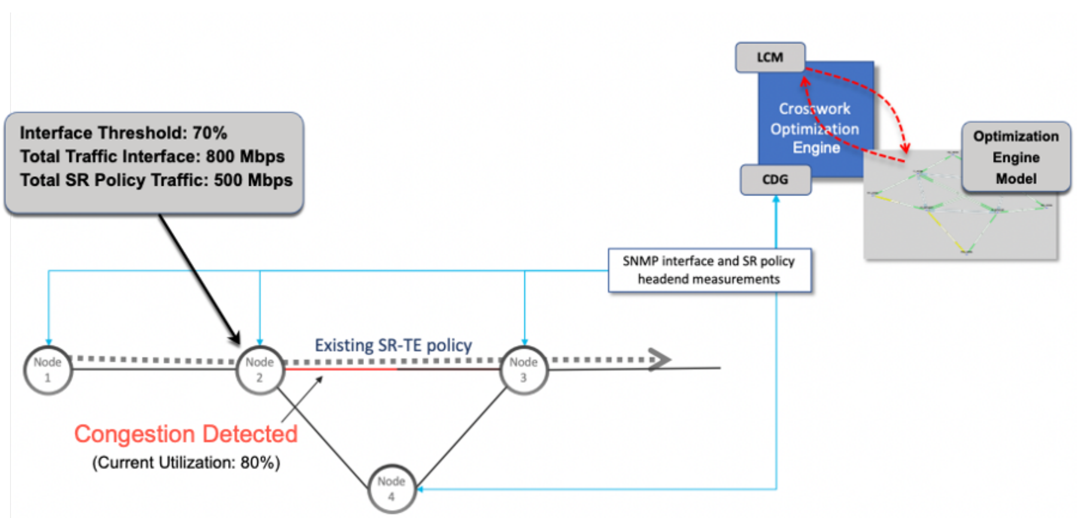
Instead of optimizing for bandwidth resources in the network by rerouting traffic in the entire network (end-to-end path optimization), LCM checks the capacity locally, in and around the congested area, at an interface level and reroutes traffic between the endpoints of the congested interface (local interface-level optimization). Focusing on an issue locally eliminates the need for simulating edge-to-edge traffic flows in the network through a full traffic matrix, which is both cumbersome and less scalable as node counts continue to increase.

When congestion is detected in the network, LCM provides recommendations to divert the minimum amount of traffic away from the congested interface. LCM collects the SR-TE policy and interface counters through SNMP. It estimates the amount of traffic that may be diverted and, if the user approves, performs the mitigation through the deployment of Tactical Traffic Engineering (TTE) SR-TE policies. Mitigating congestion locally does not require the use of the full Segment Routing Traffic Matrix (SR-TM). TTE SR-TE policies are created at the device on only either side of the congested link, with the shortest paths possible that do not congest interfaces elsewhere.

### How does LCM work?

1. First, network operators create domains that define "local" portions of the network. A domain can be the entire network, but more commonly, it matches one or more geographical areas or groups of device interfaces. In this example, we have defined a domain with four devices and all their interfaces. We also assume that all the links in this domain are 1Gpbs.
2. Operator specifies a threshold defining what "congestion" means for a particular domain. In this example, the operator has set the domain's congestion threshold to 70%. The congestion threshold you decide on may vary. For guidance on how to determine what congestion threshold is best for your network and its domain architecture, see Cisco's Local Congestion Mitigation (LCM) White Paper.
3. LCM first analyzes the Optimization Engine Model (a real-time representation of the physical network, topology, and traffic) on a regular cadence. After a congestion check interval, LCM detects congestion when Node 2 utilization goes above the 70% utilization threshold.

*Figure 74: How does LCM work?*




4. LCM calculates how much traffic is eligible to divert. LCM will follow these rules and restrictions in its recommendations:

LCM only diverts traffic that is not already routed by an existing SR policy (for example, unlabeled, IGP-routed, or carried via FlexAlgo-0 SIDs). Traffic within an SR policy will not be included in the LCM calculation and will continue to travel over the original programmed path.

LCM computes diversion-eligible traffic by taking the interface traffic statistics for all traffic on the interface and subtracting the sum of traffic statistics for all SR-TE policies that flow over the interface.

*Total interface traffic – SR policy traffic = Eligible traffic that can be optimized*

This process must account for any ECMP splitting of SR policies to ensure the proper accounting of SR policy traffic. In this example, the total traffic on congested Node 2 is 800 Mbps and the total traffic of all SR policies routed over Node 2 is 500 Mbps.

The total traffic that LCM can divert in this example is 240 Mbps. That is, 800 Mbps – 560 Mbps = 240 Mbps

5. LCM calculates the amount of traffic that must be sent over alternate paths by subtracting the threshold-equivalent traffic from the total traffic on the interface. In this example, the amount to be diverted is 100 Mbps:

   800 Mbps – 640 Mbps (70% threshold) = 100 Mbps

   LCM must route 100 Mbps of 300 Mbps (eligible traffic) to another path.

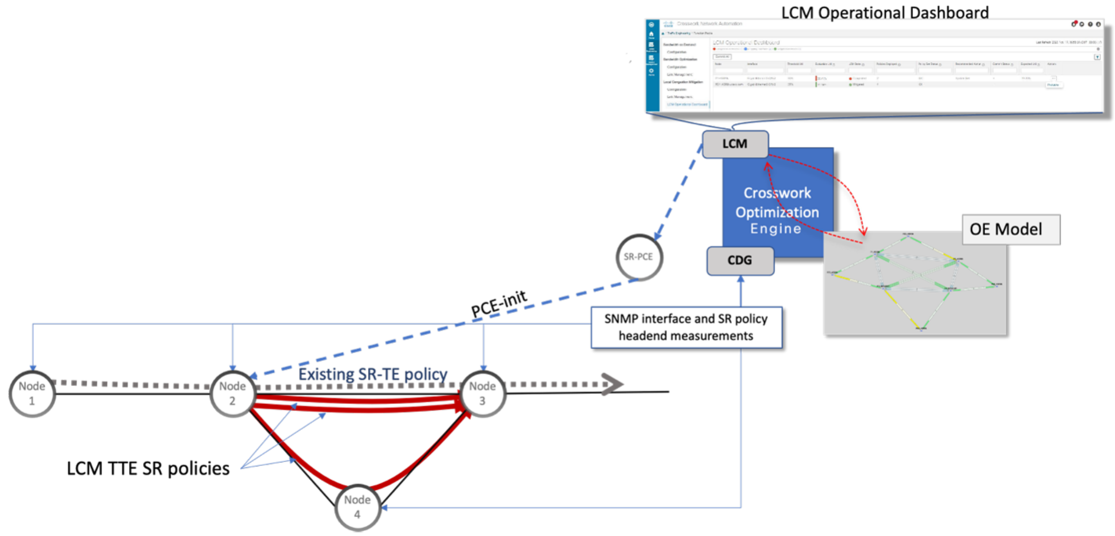
6. LCM determines the number of TTE SR policies are needed and their paths. The ratio of how much LCM eligible traffic can stay on the shortest path to the amount that must be rerouted, will determine the number of TTE SR policies needed on the shortest and alternate paths, respectively.

   In this example, LCM must divert one-third of the total eligible traffic (100 Mbps out of 300 Mbps) away from the congested link. Assuming a perfect ECMP, LCM estimates three tactical SR-TE policies are required to create this traffic split: one tactical SR-TE policy will take the diversion path, and two tactical SR-TE policies will take the original path. There is sufficient capacity in the path between Node 2 and Node 4. Therefore, LCM recommends three TTE SR policies (each expected to route approximately 100 Mbps) to be deployed from Node 2 to Node 3 via SR-PCE:

   - 2 TTE SR policies to take a direct path to Node 3 (200 Mbps)
   - 1 TTE SR policy takes a path via Node 4 (100 Mbps)

   These recommendations will be listed in the **LCM Operational Dashboard**.

*Figure 75: Recommended policies on the LCM operational dashboard*




Assuming you deploy these TTE SR policies, LCM continues to monitor the deployed TTE policies and will recommend modifications or deletions as needed in the LCM **Operational Dashboard**. TTE SR policy removal recommendations will occur if the mitigated interface is not congested if these policies are removed (minus a hold margin). This helps to avoid unnecessary TTE SR policy churn throughout the LCM operation.

### Circuit-Style policies

Circuit-Style Segment Routing Policies (CS-SR or CS policies) are connection-oriented transport services that you can use to implement what are sometimes referred to as "circuit emulations" or "private lines". Combining segment-routing architecture's adjacency SIDs with stateful PCEP path computation, CS policies provide:

- Persistent, dedicated, bi-directional, co-routed transport paths with predictable latencies and other performance metrics in both directions.
- Guaranteed bandwidth commitments for traffic-engineered services using these paths.
- End-to-end path protection to ensure there is no impact on Service Level Agreements.
- Automatic monitoring, maintenance, and restoration of path integrity.
- Flexible operations, administration, and management of Circuit-Style paths.
- A software-defined replacement for older CEM infrastructure, such as SONET/SDH.

### How do Circuit-Style policies work?

The initial configuration of CS policies follows these steps:

1. Crosswork Network Controller and its applications discover and map the network topology.
2. Crosswork users enable CS policy support, specifying the base bandwidth to be allocated to CS policies as a whole and a threshold percentage of bandwidth usage that will generate an alarm when exceeded on any CS-calculated path. So, for example, on a 1 GB link with 20 percent of bandwidth reserved for Circuit Style use, CS policies can use up to 200 Mbps of that link. Note, however, that if the bandwidth minimum

threshold is set to the default of 80 percent, alarms will be generated as soon as 160 Mbps of the link is used.

3. Network operators create a CS policy for each set of nodes for which they want to establish a guaranteed path. The policy specifies the two nodes to be linked by the main path, the bandwidth to be reserved, and the backup path. To accommodate bandwidth and path failures, the configuration must include bi-directionality, path protection, and performance-management liveness-detection settings.
4. When the operator commits the CS policy, the device-resident Path Computation Client (PCC) will request the Crosswork-resident PCE server to compute candidate Working and Protected paths that conform to the CS policy's bandwidth and other constraints (using a single PCEP request message).
5. When CS policy support is enabled, the PCC computes both paths and deducts their CS policy-guaranteed bandwidth from the allocated available bandwidth.
6. Crosswork replies to the PCC with the primary Working and Protected path lists and commits to, or "delegates," them. The topology map displays the current Active and Protected paths between the two nodes, using the colors configured when the CS policy was configured, and labels the two endpoint nodes so they can be identified as CS policy endpoints.

After the initial configuration:

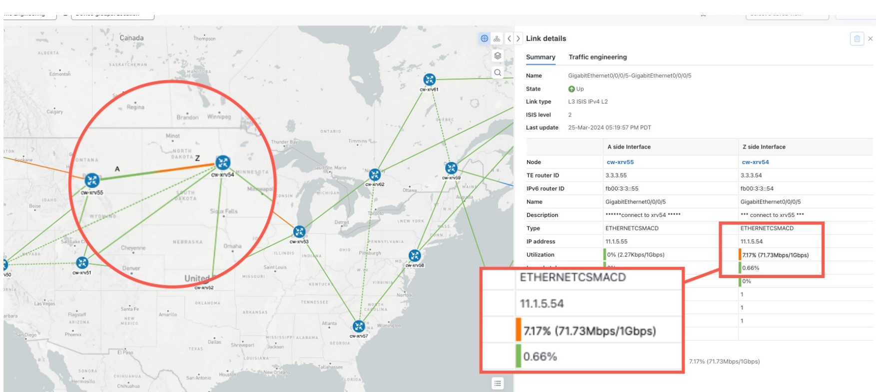
1. Crosswork monitors the delegated path and the active CS policies. It updates the available and reservable bandwidth in the network in near real-time.
2. Crosswork generates threshold-crossing alarms when bandwidth usage or additional CS policy requirements exceed the configured reserved bandwidth or bandwidth usage threshold.
3. If delegated paths fail for any reason, Crosswork recomputes paths as needed.

# Scenario: Use LCM to reroute traffic on an overused link

In this scenario, we will enable Local Congestion Mitigation (LCM) and observe its congestion mitigation recommendations. LCM will recommend that we deploy Tactical Traffic Engineering Segment Routing (TTE SR) policies on a device's interfaces when usage exceeds a defined threshold. We will preview the recommended TTE SR policies before committing them.

This example uses the following topology:

***Figure 76: Initial utilization***



> **Note** If you are viewing the HTML version of this guide, click on the images to view them in full-size.

We will enable LCM with a configuration that results in the link between **cw-xrv54** and **cw-xrv55** becoming over-used. We will then review the mitigation solutions Crosswork calculates. In this example, the operator decides whether to apply the solution.

# Assumptions and prerequisites

The following sections list high-level requirements that must be met to ensure proper LCM operation.

### Congestion evaluation requirements

LCM requires traffic statistics from the following:

- Interface traffic measurements
- Headend SR-TE policy traffic measurements

To ensure LCM is receiving these traffic statistics:

- Enable SNMP on the devices whose traffic you want to monitor, including the headend device. For more on this task, see Configuring SNMP Support. Note that gNMI is also an option for collecting traffic measurements.
- Ensure that the SNMP-enabled devices are all reachable from the Crosswork Data Gateway. For more on this task, see Check Connectivity to the Destination.
- Configure the headend device to use strict SID labels for SR policies. To perform this task:

  1. Enable segment routing on the headend device and configure the segment routing global block (SRGB) and the segment routing local block (SRLB) ranges. For example:

     ```
     segment-routing
      mpls
       global-block 16000 23999
       node-msd 16
      !
      srlb 15000 15999
     ```

  2. Configure the SR policy candidate paths to use strict SID labels. You can use either explicit paths or dynamic paths with constraints. For example:

     ```
     segment-routing
      traffic-eng
       policy COLOR-100-TO-10.0.0.1
        color 100 end-point ipv4 10.0.0.1
        candidate-paths
         preference 100
          explicit segment-list SL1
         !
         preference 200
          dynamic
           constraints
            affinity include-any RED BLUE
            sid-algorithm strict-spf
           !
          !
     ```

```
    !
   !
  !
 !
!
segment-list SL1
 index 10 mpls label 16001 node 10.0.0.2 strict
 index 20 mpls label 16002 node 10.0.0.3 strict
 index 30 mpls label 16003 node 10.0.0.4 strict
!
```

3. Configure the SR policy headend behavior using the binding SID and the autoroute announce option. For example:

```
!segment-routing
traffic-eng
pcc
profile 1
autoroute
include ipv4 all
force-sr-include
!
!
!
!
```

### Congestion mitigation requirements

The headend device must support PCE-initiated SR-TE policies with autoroute steering. However, LCM will not work if the headend is a Cisco NCS device with L2VPN traffic in the network.

Devices should be configured with **`force-sr-include`** to enable traffic steering into SR-TE policies with autoroute. For example:

`segment-routing traffic-eng pcc profile ID autoroute force-sr-include`

The *`ID`* parameter in this command identifies the PCC profile associated with the SR-TE policy that PCE has provisioned. The ID value can be any integer from 1 to 65535, but it must match the profile ID that PCE uses to instantiate the policy. If not, the policy will not be activated. For example, suppose PCE provisions a policy with profile ID `10`. In that case, you must configure `segment-routing traffic-eng pcc profile 10 autoroute force-sr-include` on the headend router to enable autoroute announcement for that policy. For more information, see the *Segment Routing Configuration Guide, Cisco IOS XE 17 (Cisco ASR 920 Series)*, COE-PCE Initiated SR Policy with OSPF and IS-IS SR-TE Autoroute Announce.

**Note** The ID that is configured under the PCC profile, must match the Profile ID option set in the LCM Configuration page.

The headend device must support Equal Cost Multi-Path (ECMP) across multiple parallel SR-TE policies. To verify that a device can support SR-TE policies using ECMP, check that the device has the following:

- Segment Routing is enabled and configured with a Segment Routing Global Block (SRGB) that matches the SRGB of the SR-TE policy headend and tailend routers. Use the `show segment-routing mpls state` command to verify the SRGB configuration on the device.
- BGP-LS is enabled and configured to advertise and receive link-state information from the SR-TE policy headend and tailend routers. Use the `show bgp link-state link-state` command to verify the BGP-LS

status and the `show bgp link-state link-state database` command to verify the link-state information on the device.

- ECMP is enabled and configured to load-balance traffic across multiple equal-cost paths based on flows. Use the `show ip route` command to verify the ECMP routes and the `show ip cef` command to verify the ECMP load-balancing algorithm on the device.

If all these conditions are met, then the device can support an SR-TE policy using ECMP.

### Related topics

For more information and examples on how to configure and verify SR-TE policies, see:

- Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers
- Segment Routing Configuration Guide, Cisco IOS XE 17 | Access and Edge Routers

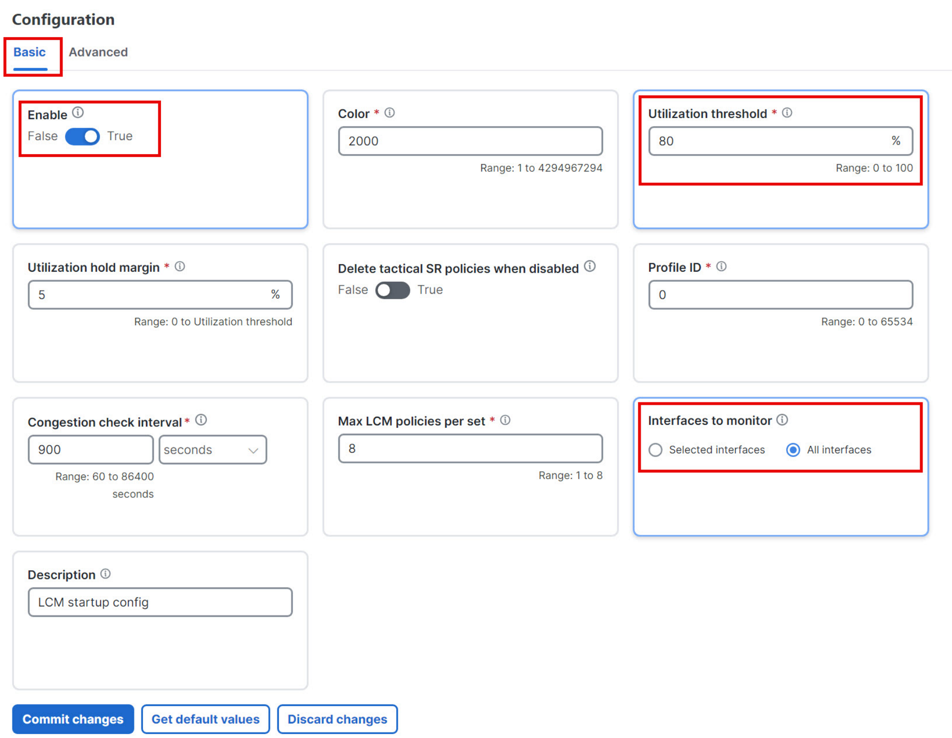
# Step 1: Enable LCM and configure the utilization thresholds

In this step, enable LCM and configure the global utilization threshold.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Local Congestion Mitigation** > **Domain-ID** and click **Configure**.

**Step 2** Toggle the **Enable** switch to **`True`**, and enter the global utilization threshold you want to set. In this case, we set the threshold at 80%, and select the **Interfaces to monitor** > **All Interfaces** option. To see information about other options for each configuration setting, hover the mouse over **i** (help icon).

*Figure 77: Basic LCM configuration*




**Step 3** Click **Commit Changes**.

**Note** After committing the configuration changes, LCM will display *recommendations* on the **LCM Operational Dashboard** if congestion occurs on any monitored interfaces. LCM will *not* commit or deploy new TTE policies automatically. Later, you will be able to preview the recommended TTE policies and decide whether or not to commit and deploy them onto your network.

**Step 4** You can also define individual interface thresholds. Go to the **Interface Thresholds** page (**Traffic Engineering** > **Local Congestion Mitigation** > ***Domain-ID*** >  > **Interface Thresholds**). You can add interfaces individually or upload a CSV file with a list of nodes and interfaces with custom utilization thresholds. For more information, see Add Individual Interface Thresholds.

See the following example and note the defined threshold for cw-xrv54 with interface GigabitEthernet0/0/0/1 is 20%.

**Note** The utilization thresholds in this example are extremely low and best used for lab environments.

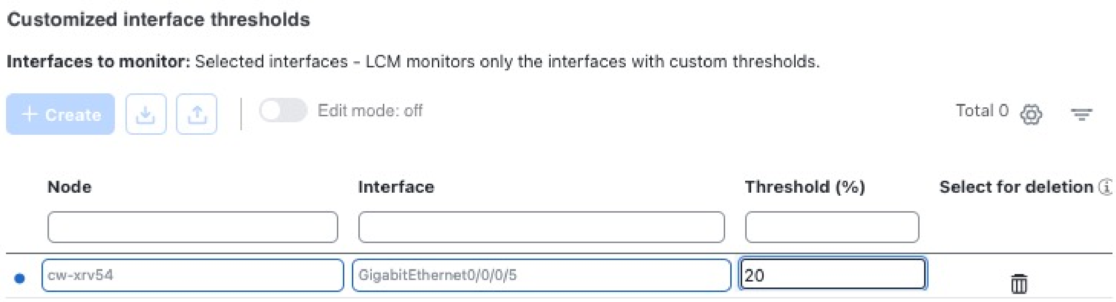



*Figure 78: Customized interface thresholds*

# Step 2: View link congestion on the map

The link between **cw-xrv60** and **cw-xrv62** is now congested.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Traffic Engineering**.

**Step 2** Click on the link to view link details, including utilization information. Usage has surpassed the custom LCM threshold defined at 20% for node **cw-xrv54** with interface GigabitEthernet0/0/0/5.

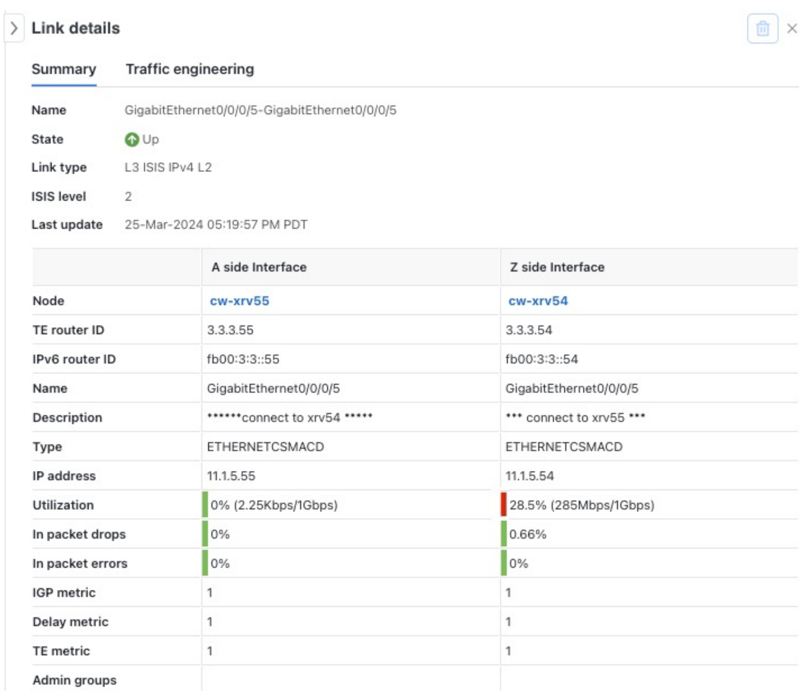



*Figure 79: Link congestion on the map*

# Step 3: View TTE SR policy recommendations in the LCM operational dashboard

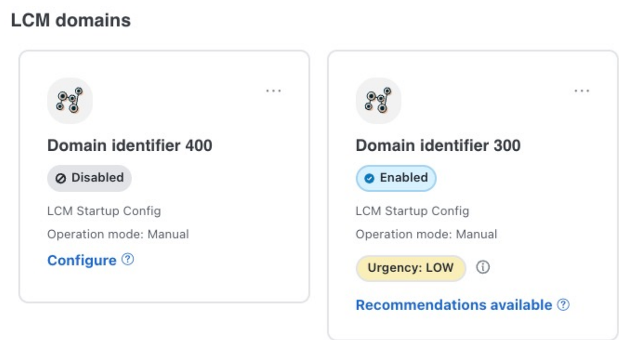
LCM has detected the congestion and computed tactical policies to mitigate it. We can preview these policies and then decide whether or not to commit them.

In this scenario, the congested device is healthy, reachable, and in sync with Crosswork. The actions we take and policies we implement will be different if, in addition to congestion, the device is down, unreachable, or out of sync.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Local Congestion Mitigation**.

When congestion is detected, the domain displays the urgency type and recommendations that are available. Click the question mark icons to display more information about the urgency type and when the most recent recommendation was given.

*Figure 80: Congested detected and LCM recommendations*

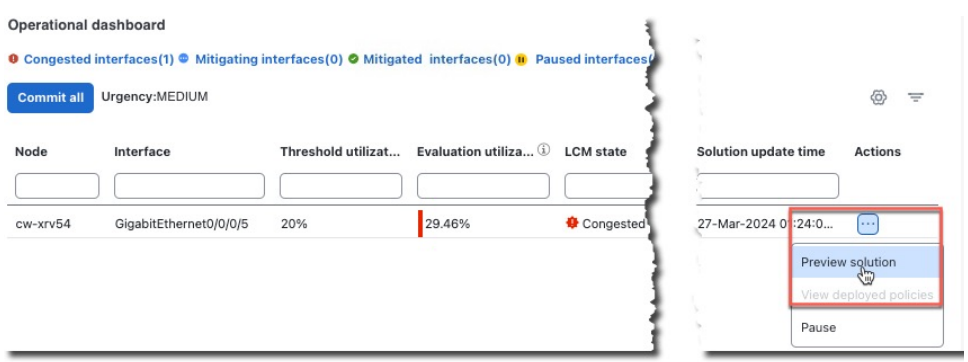



**Step 2** Open the Operational Dashboard (**Services & Traffic Engineering** > **Local Congestion Mitigation** > **Domain-ID** > **...** > **Operational Dashboard**).

The dashboard shows that cw-xrv54 utilization has surpassed 20% and is now at 29.46%. In the **Recommended Action** column, LCM recommends the deployment of TTE policy solution sets (**Recommended action - Create set**) to address the congestion on the interface.

**Step 3** Before committing TTE policies, you can preview the deployment of each TTE policy solution set. Click in the **Actions** column and choose **Preview Solution**.

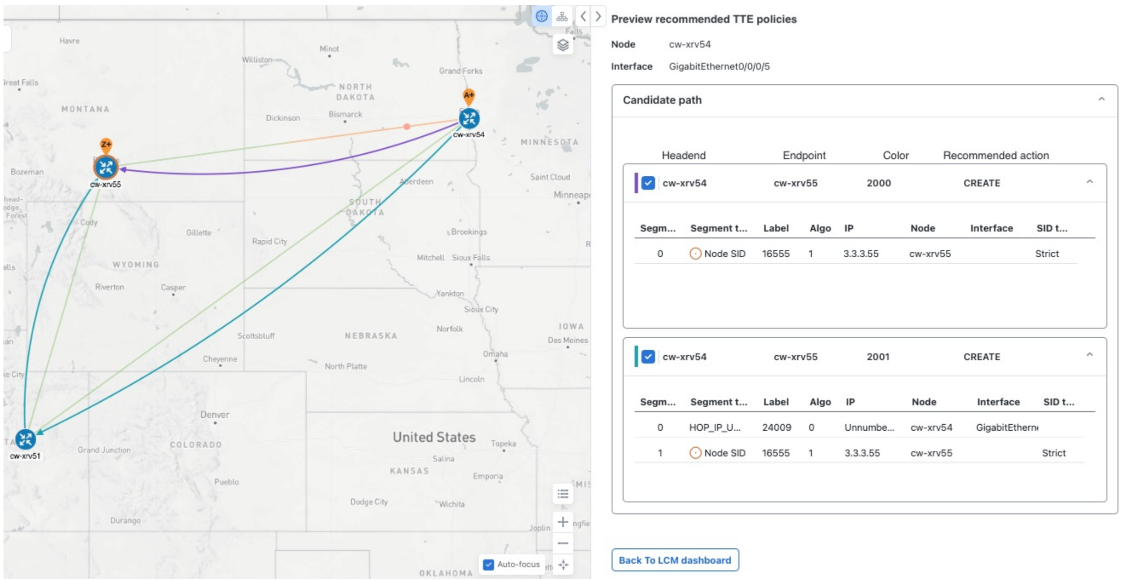
*Figure 81: Operational dashboard - preview solution*



The resulting window displays the node, interface, and the recommended action for each TTE policy. From the Preview window, you can select the individual TTE policies and view different aspects and information as you would normally on the topology map. You can expand each policy to view individual segments. After reviewing the potential implications on your network, you can decide whether or not to deploy the bypass policies that LCM recommends.

The following figure shows the recommended TTE policies for node cw-xrv54.

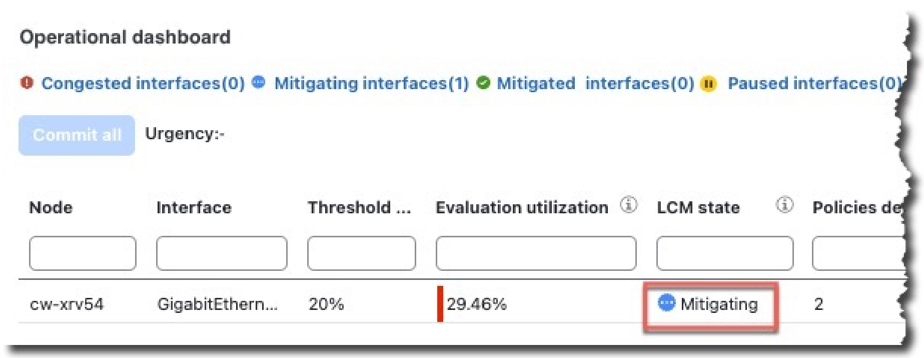
*Figure 82: Preview recommended TTE policies*




**Step 4** After you are done viewing the recommended TTE policies on the map, go back to the **Operational dashboard** and click **Commit All**. The LCM **State** column changes to **Mitigating**.

All LCM recommendations per domain must be committed to mitigate congestion and produce the expected utilization, as shown in the **Operational dashboard**. The mitigating solution is based on *all* LCM recommendations being committed because of dependencies between solution sets.

*Figure 83: Operational dashboard*



## Step 4: Validate TTE SR policy deployment

To validate the TTE SR policy deployment, follow the steps given below:

**Step 1** With the **Operational dashboard** displayed, click the at the top right of the user interface to open the **Alarms** window, then select the **Events** tab. You can use these two tabs to monitor LCM alarms and events. The **Events** tab shows the events for the LCM recommendations, the commit actions, and exceptions.

Crosswork will report network events that are detected based on the policies and features you have enabled. For example, if a link drop causes an SR-TE policy to go down or LCM detects congestion, an event is displayed in the UI.

**Step 2** Return to the **Operational dashboard** to see that the LCM state changes to **Mitigated** for all TTE policy solution sets.

**Note** The LCM state change can take up to twice as long as the SNMP cadence.

**Step 3** Confirm the TTE policy deployment by viewing the topology map. Click in the **Actions** column and choose **View Deployed Policies**. The deployed policies are displayed in focus within the topology map. All other policies are dimmed.

## Step 5: Remove TTE SR policies on LCM recommendation

After some time, the deployed TTE SR policies may no longer be needed. This occurs if utilization continues to stay under threshold without the LCM-initiated TTE policies. If this is the case, LCM generates new recommended actions to delete the TTE SR policy sets.

To remove the TTE SR policies upon LCM recommendation, follow the steps given below:

**Step 1** If needed: Display the topology map and click in the **Actions** column. Choose **View deployed policies**.

**Step 2** Click **Commit all** to remove the previously deployed TTE SR policies.

**Step 3** Confirm the removal by viewing the topology map and SR Policy table.

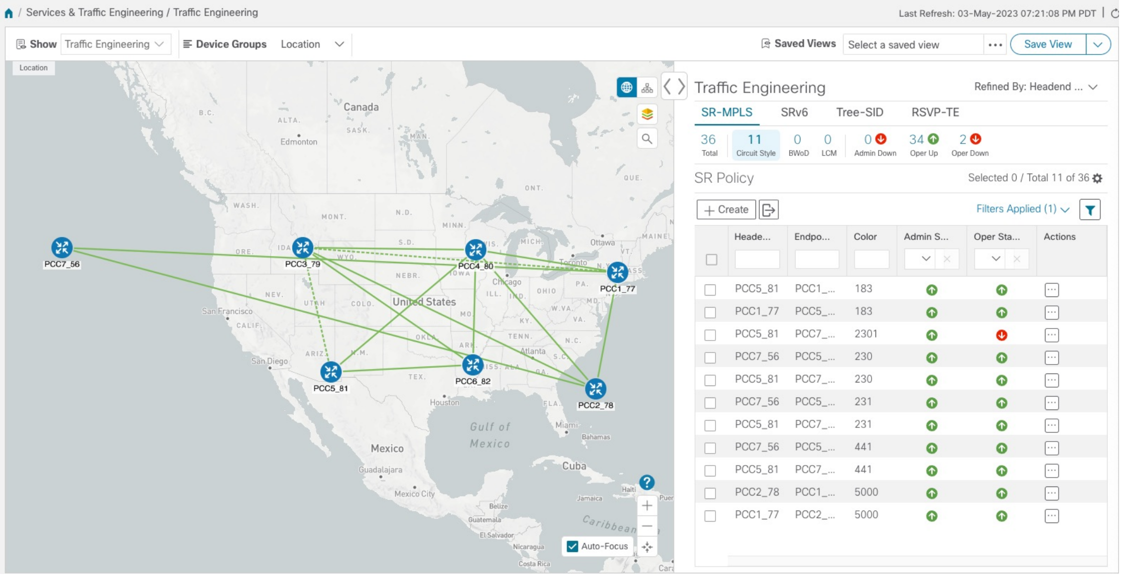
## LCM Scenario: Summary and conclusion

In this scenario, we observed how to leverage LCM to alleviate traffic congestion in the network. LCM takes the manual tracking and calculation out of your hands but, at the same time, gives you control as to whether to implement the congestion mitigation recommendations or not. You can preview the recommendations and see how the potential deployment will take effect in your network before you deploy them. As traffic changes, LCM tracks the deployed TTE SR policies and decides whether or not they are still needed. If not, LCM recommends deleting them.

# Scenario: Use CS-SR Policies to Reserve Bandwidth

In this scenario, we enable Circuit-Style Segment Routing Traffic Engineering (CS-SR, or CS SR-TE) policies and set bandwidth-reservation parameters, then configure a CS-SR policy and visualize it on the topology map. We will inspect the policy's details, including its computed Active (working) and Protected (protect) paths.

The examples in this scenario use the following topology:




We will observe what happens when the Active bandwidth-reserved path between the NCS1 and NCS3 nodes fails. We will then re-optimize the failed path.

# Assumptions and prerequisites

The following sections list high-level requirements for proper CS-SR operation, including requirements and constraints on the policy attribute values set in each CS-SR policy and the processing logic followed during path reversions.

In addition to the constraints discussed in the following sections:

- The Crosswork Circuit Style Manager (CSM) feature pack is a feature of Crosswork Network Controller Essentials. All licensed features are available during the 90-day trial period. After the trial period, you must have a license for Crosswork Network Controller Optimization Engine to continue using CSM.
- Circuit-Style policy configuration was introduced with Crosswork Network Controller 5.0. To use it, you must install version 7.9.1 (or later) of the Cisco IOS-XR Path Computation Client (PCC) on your devices. If you have been using a previous version of CNC with IOS-XR version 7.7.1 or earlier, please upgrade to version 7.9.1 or later before attempting to configure CS-SR policies.
- When using CSM with Crosswork Network Controller, the UI navigation starts from **Traffic Engineering & Services**. When using CSM with Crosswork Network Controller Optimization Engine, the navigation starts from **Traffic Engineering**.

### CS policy attribute constraints

In this scenario, we will build a CS policy between node NCS1 and node NCS 3. The policy will use the following settings and constraints:

- **PolicyName**: `NCS1-NCS3`

- **Headend Device**: `NCS1`
- **Headend IP Address**: `192.168.20.4`
- **Tailend Device**: `NCS3`
- **Tailend IP Address**: `192.168.20.14`
- **Color-choice**: `1000`
- **Bandwidth**: `10000`
- **path-protection**: Enabled
- **disjoint-path**: Enabled
- **disjoint-path forward-path type**: `Link`
- **disjoint-path forward-path group-id**: `531`
- **disjoint-path reverse-path type**: `Link`
- **disjoint-path reverse-path group-id**: `5311`
- **performance-measurement** : Enabled.
- **performance-measurement profile-type**: `Liveness`
- **performance-measurement liveness-detection**: Enabled
- **performance-measurement profile**: `CS-active`
- **working-path**: Enabled
- **working-path preference**: `100`
- **working-path dynamic-path**: Enabled
- **working-path dynamic-path pce**: Enabled
- **working-path dynamic-path metric type**: `igp`
- **working-path dynamic-path bidirectional-association-choice**: Enabled
- **working-path dynamic-path bidirectional-association-id**: `230`
- **working path dynamic constraints segments**: Enabled
- **working-path constraints segments protection**: `unprotected-only`
- **protect-path**: Enabled
- **protect-path preference**: `100`
- **protect-path dynamic-path**: Enabled
- **protect-path dynamic-path pce**: Enabled
- **protect-path dynamic-path metric type**: `igp`
- **protect-path dynamic-path bidirectional-association-choice**: Enabled
- **protect-path dynamic-path bidirectional-association-id**: `231`

- **protect-path dynamic constraints segments**: Enabled
- **protect-path constraints segments protection**: `unprotected-only`
- **restore-path**: Enabled
- **restore-path preference**: `100`
- **restore-path dynamic-path**: Enabled
- **restore-path dynamic-path pce**: Enabled
- **restore-path dynamic-path metric type**: `igp`
- **restore-path dynamic-path bidirectional-association-choice**: Enabled
- **restore-path dynamic-path bidirectional-association-id**: `232`
- **restore-path dynamic constraints segments**: Enabled
- **restore-path constraints segments protection**: `unprotected-only`

The following table shows all the options you can choose when building a policy. It is essential to understand that the attributes described in the table act as constraints. Each of them corresponds to elements of the configuration that Cisco Crosswork uses to govern how Circuit-Style path hops are computed. Each value is effectively a path computation or optimization constraint since they either specify a required property of a path or exclude possible choices for that path.

There are dependencies that must be met as well as combinations that are not allowed. The system will warn you when these sorts of issues arise. We encourage you to experiment and learn how to provision services in your network that match the types of services you want to deliver.

***Table 1: Supported Circuit Style SR-TE policy attribute values and constraints***

| Attribute | Description |
| --- | --- |
| Policy Path Protection | The path protection constraint is required for both sides of a Circuit Style SR-TE policy. |

| Attribute | Description |
|---|---|
| Bandwidth Constraint | The bandwidth constraint is required and must be the same on both sides of a Circuit Style SR-TE policy. Bandwidth changes can be made to existing policies with these effects:<br>• Once you configure the new bandwidth on both sides, Crosswork will evaluate the path. This **will not** result in a recomputed path.<br>• If the new bandwidth is higher, Crosswork checks the existing path to ensure sufficient resources. Suppose all currently delegated paths can accommodate the new bandwidth. In that case, Crosswork returns the same path with the new bandwidth value, indicating to the path computation client (PCC) that it was successful. If any current paths cannot accommodate the new bandwidth, it returns the old bandwidth value, indicating that it was unsuccessful. This evaluation will not be retried unless the bandwidth is changed again.<br>• If the bandwidth is lower, Crosswork returns the same path with the new bandwidth value to indicate to the PCC that it was successful.<br>The user interface shows both the requested and reserved bandwidth under each candidate path when you view the policy details. These values can differ if the requested bandwidth is increased but there is insufficient available CS pool bandwidth along one or more of the paths. |
| Candidate Paths and Roles | The `Working` path is defined as the highest preference Candidate Path (CP).<br>The `Protect` path is defined as the CP with the second highest preference.<br>The `Restore` path is defined with the lowest preference CP. The headend must have `backup-ineligible` configured.<br>CPs of the same role in each direction must have the same CP preference. |
| Bi-Directional | All paths must be configured as co-routed.<br>Paths with the same role on both sides must have the same globally unique bi-directional association ID. |
| Disjointness | Working and Protect paths on the same PCC must be configured with a disjointness constraint using the same disjoint association ID and disjointness type.<br>The disjointness association ID for a Working and Protect path pair in one direction must be unique compared to the corresponding pair in the opposite direction.<br>Only the `Node` and `Link` disjoint types are supported. The disjoint type must be the same in both directions of the same policy.<br>The Restore path must not have a disjointness constraint set.<br>Crosswork follows strict fallback behavior for all Working and Protect path disjointness computations. This means that if node type disjointness is configured but no path is available, Crosswork makes no automatic attempt to compute a less restrictive link type disjoint path. |

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>Metric Type</td><td>Only the <code>TE</code>, <code>IGP</code>, and <code>Latency</code> metric types are supported. The metric type used must match across Working, Protect, and Restore paths in both directions.</td></tr>
<tr><td>Segment Constraints</td><td>All Working, Protect, and Restore paths must have the following segment constraints:<ul><li><code>protection unprotected-only</code></li><li><code>adjacency-sid-only</code></li></ul>To ensure persistence through link failures, configure static adjacency SIDs on all interfaces that might be used by Circuit Style policies.</td></tr>
<tr><td>Supported Policy Changes</td><td>The following constraints may be changed for an operationally "up" Circuit Style SR-TE policy that has been previously delegated:<ul><li>Metric type</li><li>Disjoint type</li><li>MSD</li><li>Affinities</li></ul>Once configuration changes are consistent across all CPs and both PCCs (for example, the new metric type is the same for all CPs and both sides), Crosswork will initiate a recompute, which can result in new Working, Protect, and Restore paths.<br><br>During any transitory period in which configurations are not in sync between paths on the same PCC or between PCCs, no path updates are sent to the PCCs.</td></tr>
<tr><td>Path Computation</td><td>Crosswork computes paths for circuit style policies only after a complete bi-directional, path-protected set of candidate paths has been delegated, including Working and Protect paths on both sides.<br><br>Crosswork computes the Restore path only after the Working and Protect paths are down. The SR Circuit Style Manager feature pack configuration interface provides a configurable delay timer to control how long to wait after Restore paths are delegated from both sides before computing the path. This delay allows topology and SR policy state changes to fully propagate to Crosswork in cases where these changes triggered the Restore path delegation.<br><br>Path computation is supported for Intra/Inter area/level and Intra/Inter IGP Domain (same AS).</td></tr>
<tr><td>Reversion Behavior</td><td>Reversion behavior is controlled by the configuration of the WTR lock timer option under the Protect and Revert paths (it is not relevant for the Working path):<ul><li>No lock configuration: Revert after a default 5-minute lock</li><li>Lock with no duration specified: No reversion</li><li>Lock duration <value>: Revert after the specified number of seconds</li></ul></td></tr>
</table>

### Unsupported CS policy options

The following table lists the CS policy options, attributes, and constraints that are not supported in this version of CSM.

*Table 2: Unsupported Circuit Style SR-TE policy options*

| Attribute | Description |
|---|---|
| Unsupported Configurations | The following configurations are not supported:<br>• Metric-bounds<br>• SID-Algo constraints<br>• Partial recovery is not supported with 7.8.x.<br>• State-sync configuration between PCEs of a high-availability pair. These are not required with Circuit Style SR-TE policies. Use of this feature may result in degraded performance.<br>• Multiple Circuit Style SR-TE policies between the same nodes with the same color but different endpoint IPs. |
| Unsupported Policy Changes | The following configuration changes to a previously delegated and operationally "up" Circuit Style SR-TE policy are not supported:<br>• CP preference<br>• Disjoint Association ID<br>• Bi-directional Association ID<br>To change these configurations for an existing policy, you must first shut down the policy on both sides, make the change (complying with restrictions as detailed above in terms of consistency), and then "no shut" the policy. |
| Unsupported Path Computation | Automatic re-optimization is not supported for paths based on changes in topology, LSP state, or any periodic event. Path computation is not supported for Inter-AS. |

### Path reversion logic

Path reversion depends on the initial state of the Working, Protect and Revert paths and the events affecting each path. The scenarios in the following table provide examples of typical reversion behavior.

*Table 3: Path reversion scenarios*

| Initial State | Events | Behavior |
|---|---|---|
| Working path is down, Protect path is up/active | Working path comes back up | 1. Working path recovers to up/standby state.<br>2. Each PCC moves the Working path to active after the WTR timer expires.<br>3. Protect path moves to up/standby. |

| Initial State | Events | Behavior |
|---|---|---|
| Working path is down, Protect path is down, Revert path is up/active | Working path comes back up, then Protect path comes back up | 1. Working path recovers and goes to up/active state<br>2. Revert path is removed<br>3. Protect path recovers and goes to up/standby |
| Working path is down, Protect path is down, Revert path is up/active | Protect path comes back up, then Working path comes back up | On side A: The Working path failure is local (the first Adj SID in the SegList is **invalid**):<br>1. Protect path recovers and goes to up/active.<br>2. Recover path is removed.<br>3. Working path recovers and goes to up/standby.<br>4. Each PCC moves the Working path to active after the WTR timer expires, Protect path goes to up/standby.<br><br>On side Z: Working path failure is remote (first Adj SID in SegList is **valid**):<br>1. Protect path recovers but is not brought up, Revert path remains up/active.<br>2. Working path recovers and goes up/active.<br>3. Revert path is removed.<br>4. Protect path goes to up/standby. |

#### What happens when path failures occur?

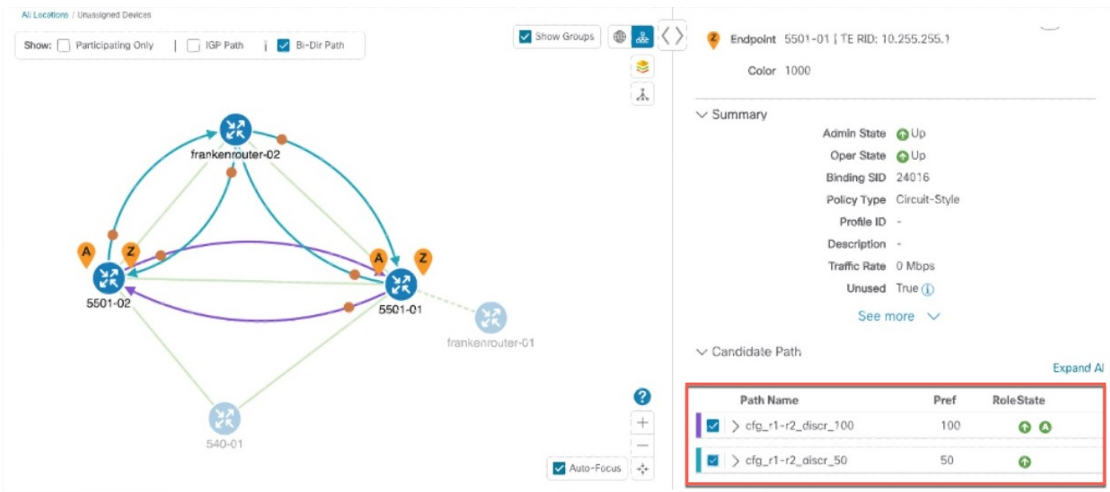
Cisco Crosswork computes paths for CS policies only after a complete bidirectional, path-protected set of candidate paths has been delegated. A path can be considered to have "failed" for various reasons, including transient changes in workloads caused by congestion elsewhere in the network or any condition that causes the path not to meet bandwidth expectations. Irrespective of the cause, three types of paths are used during these kinds of failures. Crosswork activates them as needed, according to their preference settings:

- **Working**—This is the path with the highest preference value. Crosswork always tries to keep the operational (Oper Up) path with the *highest* preference as the *Active* path.
- **Protected**—This is the path with the second highest preference. If the Working path goes down, the Protected path (with the lower preference value) is activated. After the Working path recovers, the Protected path remains active until the default lock duration expires, and then the Working path is activated.
- **Restore**—This is the path with the lowest preference path. Crosswork computes the Restore path only when the Working and Protected paths are down. You can control how long after Restore paths are delegated to wait before the path is computed. This delay allows topology and policy state changes to fully propagate through the network and allows Crosswork to gather and analyze telemetry to determine network health.

To address failures effectively and switch from the Working to the Protected path, configure Performance Measurement (PM) as part of your CS policy. For more information, see Step 4: Configure Circuit Style SR-TE policies using import, on page 125.

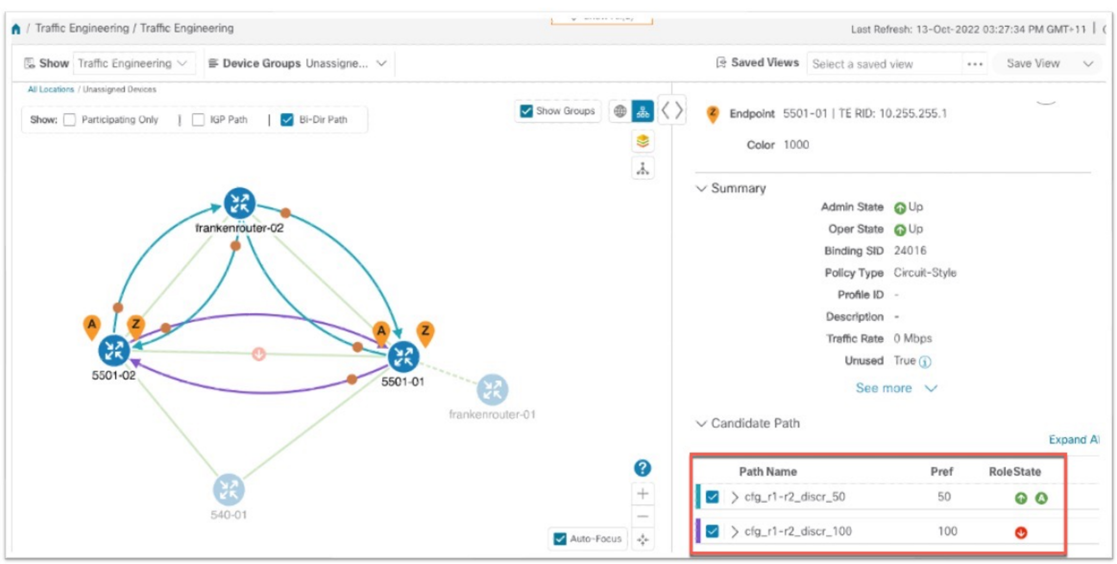
The following image shows that the Working and Protected paths of the example CS policy are operational. The *active* path is indicated by the "A" icon shown next to that path in the **State** column in the **Candidate Path** list.

*Figure 84: Initial candidate paths: working path is active*




If the Working path performance falls below expectations, the Protected path becomes Active immediately (usually under 50 milliseconds).

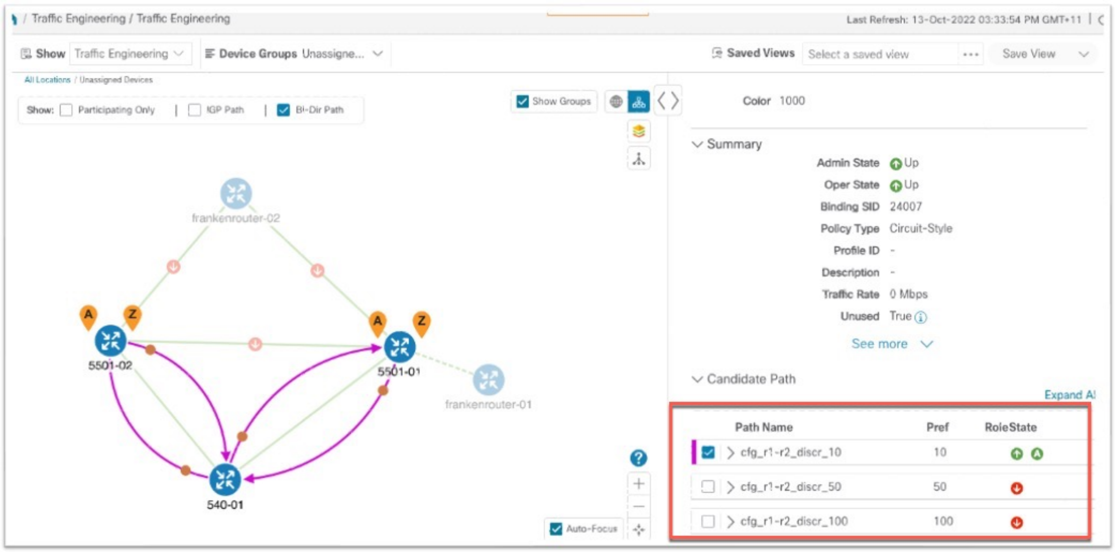
*Figure 85: Protected path becomes active*




When the Working path comes back up, the Protected path resumes the Protected role again, and the Working path (with preference 100) becomes Active again.

If both the Working and Protected paths go down, Crosswork calculates a Restore path and makes it the active path. Note that the Restore path has the lowest preference value of 10. The Restore path only appears in this particular case. If the Working or Protected paths become operational again, Crosswork will activate them, and the Restore path will disappear from the topology map and the Candidate Path list.

***Figure 86: Restore path becomes active***




# Step 1: Enable Circuit Style Manager

To manage and visualize Circuit Style SR-TE policies on the topology map, we must first enable SR Circuit Style Manager (CSM) and set bandwidth reservation settings. As soon as you define these settings, CSM computes the best bidirectional failover paths between the two nodes, while observing the requested CSM bandwidth and threshold settings, and the constraints defined in the Circuit Style SR-TE policy. The following steps show how to do this.

CSM tries to ensure that the total reserved bandwidth on all interfaces remains at or below the network-wide resource pool. When the total usage on all interfaces exceeds the threshold value you set, CSM generates a threshold-crossing alarm.

To help you estimate the Circuit Style SR-TE bandwidth pool and threshold settings that are reasonable for your organization's implementation, two examples are provided to demonstrate how CSM handles policies that exceed either the bandwidth pool size or both the pool size and alarm threshold. For this scenario, you can enter either of these examples or choose settings that are less likely to be exceeded in a practical implementation.

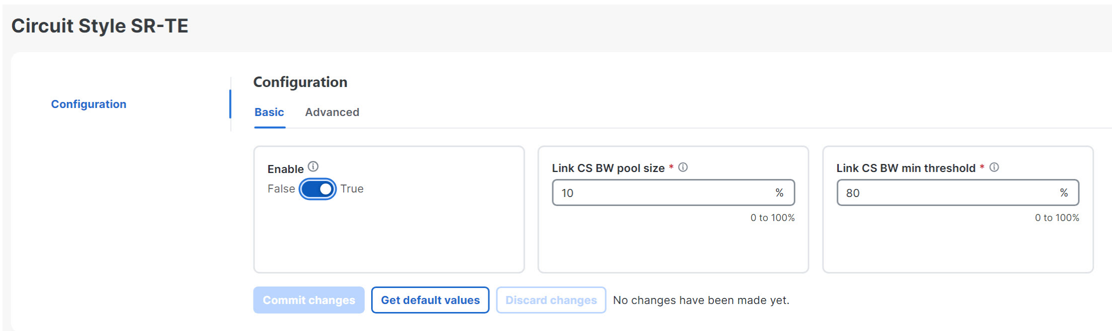
After enabling CSM, you must create Circuit Style SR-TE policy configurations. You can use any of the following methods to create Circuit Style SR-TE policies. In this scenario, we will create the same policy each time, but we will go through each method so that you can decide which methods will best meet your needs:

- Step 2: Configure Circuit Style SR-TE policies ssing device CLI
- Step 3: Configure Circuit Style SR-TE policies using add
- Step 4: Configure Circuit Style SR-TE policies using import

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

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

*Figure 87: Basic Circuit Style SR-TE configuration*



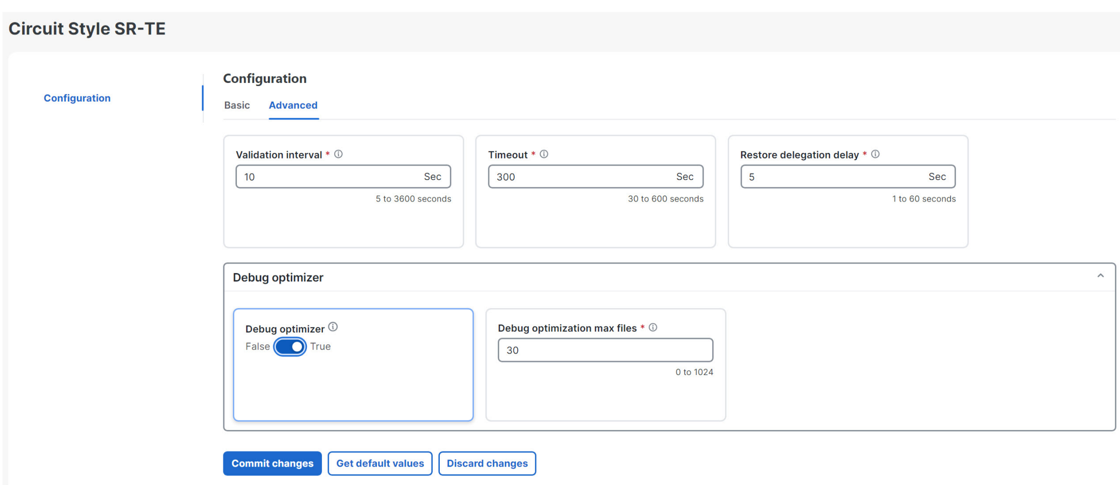

**Step 3** Enter the required bandwidth pool size and threshold information, as explained in the table below. See also the examples below, and choose one of them to enter.

| Field | Description |
|---|---|
| Link CS BW Pool Size | The percentage of each link's bandwidth reservable for Circuit Style SR-TE policies. |
| Link CS BW Min Threshold | The Link CS BW Pool utilization percentage beyond which Crosswork will generate a threshold-crossing event notification. |

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

**Step 5** (Optional): Click the **Advanced** tab to display additional CS-SR configuration values.

*Figure 88: Circuit Style SR-TE configuration - advanced tab*




a) Change the values on the **Advanced** tab as explained in the table below.

| Field | Description |
|---|---|
| Validation Interval | This is the interval that CSM will wait before the bandwidth reserved for an un-delegated policy is returned to the Circuit Style SR-TE policy bandwidth Pool. |
| Timeout | The duration CSM will wait for the delegation request before generating a threshold-crossing alarm. |
| Restore Delegation Delay | The duration CSM will pause before processing a restore path delegation. |
| Debug Optimizer | Toggle the switch to **True** to turn on the Debug Optimizer for all CS-SR policies. The Debug Optimizer will write log files to the Crosswork file system whenever it calculates routes up to the maximum number of files you specify. |
| Debug Optimization Max Files | Enter the maximum number of log files the Debug Optimizer will write. Once the maximum is reached, the Optimizer will overwrite existing files. |

b) When you are finished entering Advanced configuration values, click **Commit Changes** to save the configuration.

### Example

**Example: Bandwidth Utilization Surpasses Defined Threshold**

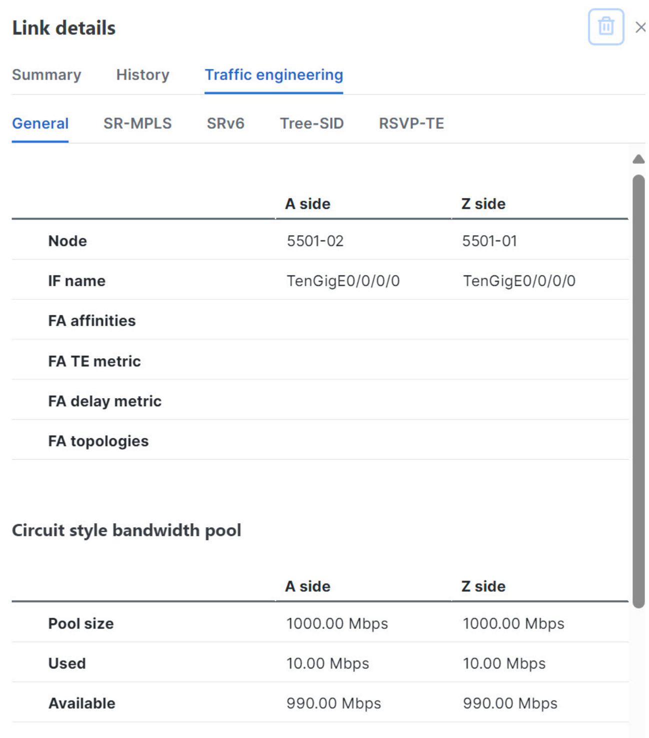
In this example, we assume the reserved bandwidth settings are as follows:

- Bandwidth Pool Size: 10%
- Bandwidth Pool Threshold: 1%

Our two nodes have 10 Gbps Ethernet interfaces, so the bandwidth pool size with these settings is 1Gbs, and the alarm threshold is 100 Mbps.

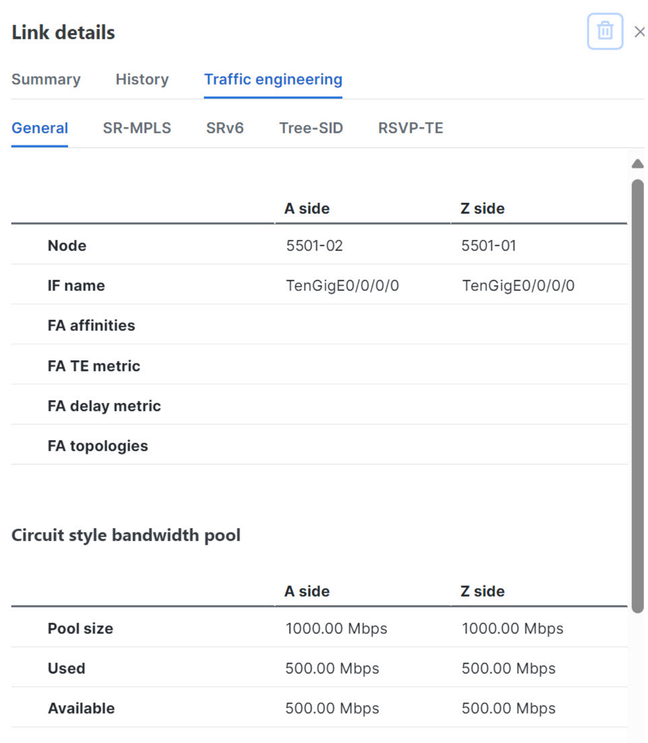
1. We create a CS policy connecting node 5501-02 to node 5501-01 (r02 - r01), with a bandwidth of 100 Mbps.

*Figure 89: CS-SR policy 10 Mbps up*



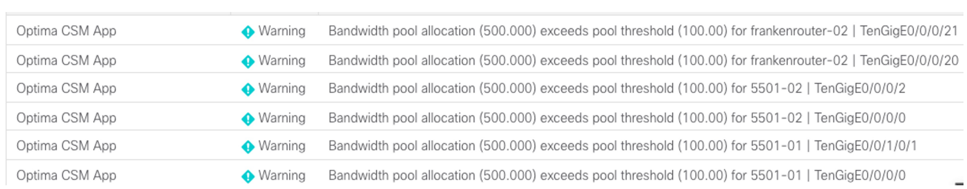

2. Later, the requested bandwidth for the policy increases to 500 Mbps. The updated CS policy is created and operational (Oper State Up).

*Figure 90: CS-SR policy 500 Mbps up*




3. Since the bandwidth utilization of 500 Mbps with the updated policy is greater than the configured bandwidth threshold (100 Mbps), Crosswork triggers alerts.

*Figure 91: Threshold alerts*




### Example: Bandwidth pool size and usage exceeded

In this example, we assume the reserved bandwidth settings are as follows:

- Bandwidth Pool Size: 10%
- Bandwidth Pool Threshold: 10%

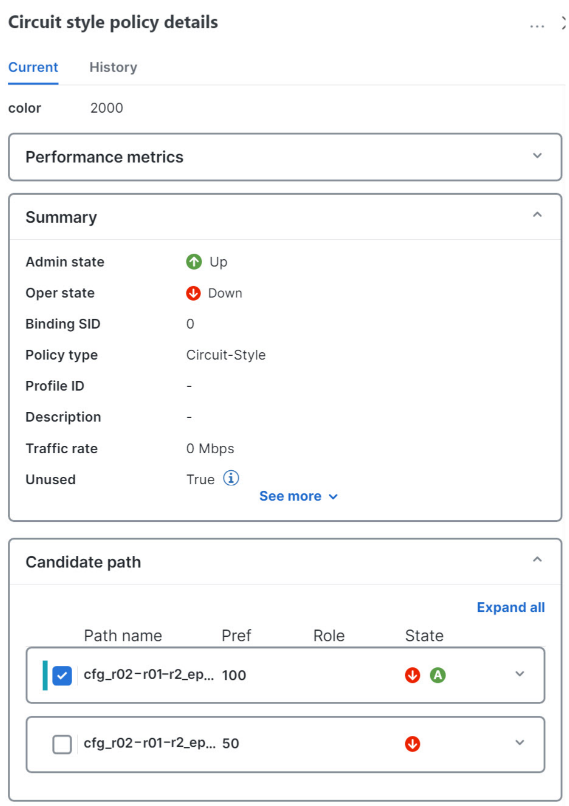
The bandwidth pool size for the 10 Gbps Ethernet interfaces is 1Gbs and the alarm threshold is 100 Mbps.

1. An existing CS-SR policy from node 5501-02 to node 5501-01 (*r02- r01*) uses a bandwidth of 500 Mbps.
2. Later, a new policy requiring a bandwidth of 750 Mbps with a path from node 5501-02 to node 5501-01 to 5501-2 (*r02- r01- r2*) is requested. Since the existing policy and this new policy

together exceed the bandwidth pool size, and alarm threshold of 1 Gbps (750 Mbps + 500 Mbps = 1250 Mbps), the following behaviors occur:

- The new CS-SR policy *r02- r01- r2* has been created but is not operational (Oper State Down).

  ***Figure 92: CS-SR policy exceeds bandwidth pool size***




- Alerts are triggered.

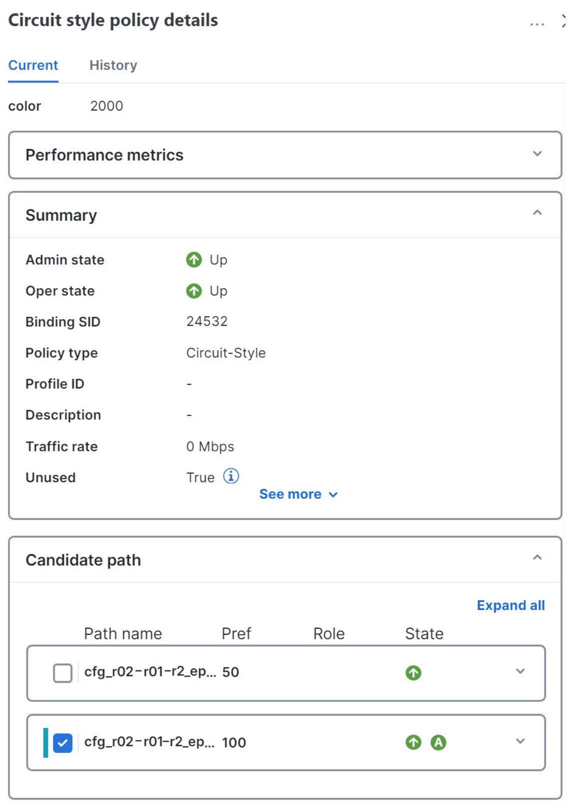
  ***Figure 93: Threshold alerts***

| Source | Severity | Description |
| --- | --- | --- |
| Optima CSM App | Warning | Unable to compute path for 10.[illegible].1 <-> 10.255.255.2 \| color 2000 due to CsmUpdateStatus.NO_PATH |
| SR Policy [10.255.255.2#10.255.... | Warning | Policy 'srte_c_2000_ep_10.[illegible].1' has operational status as DOWN. |
| SR Policy [10.255.255.1#10.255.... | Warning | Policy 'srte_c_2000_ep_10.[illegible].2' has operational status as DOWN. |

**3.** Later, the CS-SR policy *r02- r01- r2* is updated and only requires 10 Mbps. The following behaviors occur:

- Since the total bandwidth required for the two policies (10 Mbps + 500 Mbps = 510 Mbps) now requires less than the bandwidth pool size (1Gbps), CS-SR policy r02- r01- r2 becomes operational (Oper State Up).

*Figure 94: Updated CS-SR policy operational*




- Since the bandwidth utilization (10 Mbps) with the updated policy is below the configured bandwidth threshold (1 Gbps), alerts are cleared.

*Figure 95: Cleared alerts*

| Source | Severity | Description |
|---|---|---|
| SR Policy [10.[illegible].1#10.255.... | Clear | Policy 'srte_c_2000_ep_10.[illegible].2' has operational status back to UP. |
| SR Policy [10.[illegible].2#10.255.... | Clear | Policy 'srte_c_2000_ep_10.[illegible].1' has operational status back to UP. |

# Step 2: Configure Circuit Style SR-TE policies ssing device CLI

Before Cisco Crosswork, most network engineers created Circuit Style SR-TE policies directly on the devices, using the appropriate network operating system CLI commands. This step of the scenario covers direct CLI policy configuration for a Cisco device. We present it only because this is a legitimate way to create these policies, so you can see how the configuration implemented using this method matches the configuration of the other Crosswork-native methods presented in this scenario.

Crosswork Network Controller's topology discovery will automatically recognize CS policy configurations implemented directly on devices and will help you visualize them on the topology map. However, this method has some important drawbacks. To start with, you will need to be familiar with the CLI commands required to configure Circuit Style SR-TE policies properly. More importantly, Crosswork can *discover* Circuit Style SR-TE policies configured directly on a device but cannot change or delete them. We encourage you to use instead the **Add** or **Import** methods, which allow you to manage and change your configuration using Crosswork. For help using these methods, skip this step and go on to Step 3: Configure Circuit Style SR-TE policies using add or to Step 4: Configure Circuit Style SR-TE policies using import.

A Circuit Style SR-TE policy configuration must include the destination endpoint, the amount of requested bandwidth, and the bidirectional attribute. See Assumptions and prerequisites, on page 106 for additional requirements and notable constraints.

When configuring Circuit Style SR-TE policies directly on Cisco devices, make sure the configuration includes a Performance Measurement (PM) Liveness profile. A PM Liveness profile enables proper detection of candidate path liveness and effective path protection. Path Computation Clients (PCCs) do not validate past the first SID, so without PM Liveness, the path protection will not occur if the failure in the Circuit Style SR-TE policy candidate path occurs after the first hop in the segment list. Crosswork supports software-based and hardware-offload PM Liveness configuration methods. For more background on PM Liveness profiles and methods, see Configuring SR Policy Liveness Monitoring.

---

**Step 1** Use your preferred method to access the head-end device console and log in.

**Step 2** If applicable, enable the hardware module on the device for PM configuration.

**Example:**

```
hw-module profile offload 4

reload location all
```

**Step 3** Configure the Performance Measurement (PM) Liveness profile on the device. The following example uses a hardware-offload configuration.

**Example:**

```
performance-measurement
 liveness-profile sr-policy name CS-active-path
  probe
   tx-interval 3300
!
npu-offload enable   !! Required for hardware Offload only
  !
 !
 liveness-profile sr-policy name CS-protected-path
  probe
   tx-interval 3300
  !

npu-offload enable   !! Required for hardware Offload only
  !
 !
!
```

**Step 4** Configure the Circuit Style SR-TE policy. All configuration entries shown are required in order for the Crosswork CSM feature pack to manage the policy. Entry values that you must change appropriately for your network (or for your PM Liveness profile) are shown in *italics*. See Assumptions and prerequisites, on page 106 for additional requirements and notable constraints.

**Example:**

```
segment-routing
 traffic-eng
```

```
 policy NCS1-NCS3

  performance-measurement
   liveness-detection
    liveness-profile backup name CS-protected      !! Name must match liveness profile defined for
 Protect path
    liveness-profile name CS-active               !! Name must match liveness profile defined for
Active path
   !
  !
  bandwidth 10000
  color 1000 end-point ipv4 192.168.20.4
  path-protection
  ! Path protection is required on both ends of the candidate-paths
 ! Defining the Working path. Must have the highest CP preference
   preference 100
    dynamic
     pcep
     !
     metric
      type igp
     !
    !
    constraints
     segments
      protection unprotected-only
      adjacency-sid-only
     !
     disjoint-path group-id 3 type node
    !
    bidirectional
     co-routed
     association-id 230
 !
   ! Defining the Protect path. Must have second highest CP preference.
   preference 50
    dynamic
     pcep
     !
     metric
      type igp
     !
    !
    constraints
     segments
      protection unprotected-only
      adjacency-sid-only
     !
     disjoint-path group-id 3 type node
    !
    bidirectional
     co-routed
     association-id 231
    ! Defining the restore path. It must have both the lowest CP preference and backup-ineligible
setting
   preference 10
    dynamic
     pcep
     !
     metric
      type igp
     !
    !
    backup-ineligible
```

```
     !

     constraints
      segments
       protection unprotected-only
       adjacency-sid-only
      !
     !
     bidirectional
      co-routed
      association-id 232
     !
    !
    !

     !
    !

   !
  !

 !
!
```

# Step 3: Configure Circuit Style SR-TE policies using add

You can create a Circuit Style SR-TE policy between any two nodes using the Crosswork Network Controller **Add** function. This method is the simplest for users who want to be able to use Crosswork to edit or delete the Circuit Style SR-TE policies they create.

This method doesn't eliminate the need to be familiar with the CLI command attributes needed to configure Circuit Style SR-TE policies properly. If you prefer a faster method that can help you standardize these policies across your network, skip this step and use the method in Step 4: Configure Circuit Style SR-TE policies using import.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)**.

**Step 2** In the **Services/Policies** column on the left, select **SR-TE** > **Circuit-Style Policy**. Crosswork displays the **Create SR-TE > Circuit Style Policy** window.

**Step 3** Click +. Crosswork displays the **Create SR-TE** > **Circuit Style Policy** window.

**Step 4** In this scenario, we will use the name **`NCS1-NCS3`**. Enter that name in the **Name** field, then click **Continue**.

**Step 5** Begin by making the following entries in the respective fields on the **Create SR-TE > Circuit Style Policy**:

- **Name**: **`NCS1-NCS3`**
- **Color-choice**: `1000`
- **Bandwidth**: `10000`
- **path-protection**: Check the checkbox.

**Note** The color-choice and bandwidth values shown here are examples only. If you decide to follow this example in your network, use a color-choice value that is not already in use and a bandwidth value that is available within the percentage you are dedicating to CS policies.

**Step 6** Continue the scenario by entering the Circuit Style SR-TE policy constraints and specifications shown in the table below. The user interface for the **Add** function groups policy fields into related categories. Click on the field group name or the > icon at the right to expand a category and display its dependent fields.

You must change the device names and IP addresses you enter to match the devices on your network.

*Table 4: Example: Circuit Style SR-TE policy ssing add*

| Expand this: | To specify this: |
|---|---|
| head-end | • **Device**: Enter `NCS1`.<br>• **Ip-address**: Enter `192.168.20.4`. |
| tail-end | • **Device**: Enter `NCS3`.<br>• **Ip-address**: Enter `192.168.20.14`. |
| disjoint-path | Click **Enable disjoint-path**. |
| disjoint-path > forward-path | • **Type**: Select `Link`.<br>• **group-id**: Enter `531`. |
| disjoint-path > reverse-path | • **Type**: Select `Link`.<br>• **group-id**: Enter `5311`. |
| performance-measurement | Click **Enable performance-measurement**. |
| performance-measurement > Profile-type | Click **liveness**. |
| performance-measurement > Profile-type > liveness-detection | Click **Enable liveness-detection**. Then:<br>• **Profile**: Enter `CS-active`.<br>• **Backup**: Enter `CS-protected`. |
| working-path | Click **Enable working-path**. Then select **dynamic-path**. |
| working path > dynamic | Click **Enable dynamic-path**. Then:<br>• **pce**: Check the checkbox.<br>• **Metric-type**: Select `igp`<br>• **Bidirectional-association-choice**: Select **bidirectional-association-id** and enter `230`in the field. |
| working path > dynamic > constraints > segments | Click **Enable segments**. Then in the **Protection** field, select `unprotected-only`. |
| protect-path | Click **Enable protect-path**. Then select **dynamic-path**. |

| Expand this: | To specify this: |
| --- | --- |
| protect-path > dynamic | Click **Enable dynamic**. Then:<br>• **pce**: Check the checkbox.<br>• **Metric-type**: Select `igp`<br>• **Bidirectional-association-choice**: Select **bidirectional-association-id** and enter `231`in the field. |
| protect-path > dynamic > constraints > segments | Click **Enable segments**. Then in the **Protection**, field, select `unprotected-only`. |
| restore-path | Click **Enable restore-path**. Then select **dynamic-path**. |
| restore-path > dynamic | Click **Enable dynamic-path**. Then:<br>• **pce**: Check the checkbox.<br>• **Metric-type**: Select `igp`<br>• **Bidirectional-association-choice**: Select **bidirectional-association-id** and enter `232`in the field. |
| restore-path > dynamic > constraints > segments | Click **Enable segments**. Then in the **Protection** field, select `unprotected-only`. |

**Step 7** When you finish, click **Dry Run** to validate and save your changes. Crosswork will display your changes in a popup window.

If you want to configure a service with requirements that do not match those described in this example, contact Cisco Customer Experience.

**Step 8** When you are ready to activate the policy, click **Commit Changes**.

# Step 4: Configure Circuit Style SR-TE policies using import

If your organization has already implemented Circuit Style SR-TE policies and wants to roll them out on more network devices, the easiest way to do so is using Crosswork Network Controller's **Import** function. You can use **Import** to download a policy template file from Crosswork. The template file will be in JSON or XML format. You can save the template under a new name, insert the policy values of your choice, and then import the modified file.

Using the **Import** function is fast and a good way to standardize Circuit Style SR-TE policies across your network. You can set the same template files to establish CS-SR policies between multiple pairs of devices, varying only the endpoint names and addresses and any other values as appropriate for each circuit.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)**.

**Step 2** In the **Services/Policies** column on the left, select **SR-TE** > **Circuit-Style Policy**.

**Step 3** Click . Then click the **Download sample JSON and XML files (.zip)** link. The downloaded ZIP file contains templates for all the Crosswork service types, including Circuit-Style, in JSON and XML formats.

**Step 4** Unzip `samplePayload.zip` and locate the `CS-Policy.json` and `CS-Policy.xml` template files.

**Step 5** Using a JSON or XML file editor of your choice, open the `CS-Policy` template file and save it under the name **`cs1-cs4`**.

**Step 6** If using the JSON template file, edit it to look like the example below. If you are using the XML template, go on to the next step.

**Example:**

**CS-SR policy in JSON**


```
{
  "name": "NCS1-NCS3",
  "head-end": {
    "device": "NCS1",
    "ip-address": "192.168.20.4"
  },
  "tail-end": {
    "device": "NCS3",
    "ip-address": "192.168.20.14"
},
  "color": 1000,
  "bandwidth": 10000,
  "disjoint-path": {
    "forward-path": {
      "type": "Link",
      "group-id": 531
    },
    "reverse-path": {
      "type": "Link",
      "group-id": 5311
    }
  },
  "performance-measurement": {
    "profile-type": "liveness",{
      "profile": "CS-active",
      "backup": "CS-protected"
    },
  },
  "path-protection": {},
  "working-path": {
    "dynamic": {
      "constraints": {
        "segments": {
          "protection": "unprotected-only"
        }
      },
      "pce": {},
      "metric-type": "igp",
      "bidirectional-association-id": 230
    }
  },
  "protect-path": {
    "dynamic": {
      "constraints": {
        "segments": {
          "protection": "unprotected-only"
        }
      },
      "pce": {},
      "metric-type": "igp",
      "bidirectional-association-id": 231
```

```
      },
      "revertive": true
    },
    "restore-path": {
      "dynamic": {
        "constraints": {
          "segments": {
            "protection": "unprotected-only"
          }
        },
        "pce": {},
        "metric-type": "igp",
        "bidirectional-association-id": 232
      }
    }
  }
```

**Step 7** If you are using the XML template file, edit it to look like the example below.

**Example:**

**CS-SR policy in XML**

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <cs-sr-te-policy xmlns="http://cisco.com/ns/nso/cfp/cisco-cs-sr-te-cfp">
    <name>NCS1-NCS3</name>
    <head-end>
      <device>cs1</device>
      <ip-address>192.168.20.4</ip-address>
    </head-end>
    <tail-end>
      <device>cs4<device>
      <ip-address>192.168.20.14<ip-address>
    <tail-end>
    <color>1000</color>
    <bandwidth>10000<bandwidth>
    <disjoint-path>
      <forward-path>
        <type>Link</type>
        <group-id>531</group-id>
      </forward-path>
      <reverse-path>
        <type>Link</type>
        <group-id>5311</group-id>
      </reverse-path>
    </disjoint-path>
    <performance-measurement>
      <profile-type>liveness
        <profile>CS-active</profile>
        <backup>CS-protected"</backup>
      </profile-type>
    </performance-measurement>
    <path-protection></path-protection>
    <working-path>
      <dynamic>
        <constraints>
          <segments>{
            <protection>unprotected-only</protection>
          </segments>{
        </constraints>{
          <pce></pce>
          <metric-type>igp</metric-type>
          <bidirectional-association-id>230</bidirectional-association-id>
      </dynamic>
    </working-path>
```

```
      <protect-path>
        <dynamic>
          <constraints>
            <segments>
              <protection>unprotected-only</protection>
            </segments>
          </constraints>
          <pce></pce>
          <metric-type>igp</metric-type>
          <bidirectional-association-id>231</bidirectional-association-id>
        </dynamic>
      </protect-path>
    <restore-path>
      <dynamic>
        <constraints>
          <segments>
            <protection>unprotected-only</protection>
            </segments>
          </constraints>
          <pce></pce>
          <metric-type>igp</metric-type>
          <bidirectional-association-id>232</bidirectional-association-id>
        </dynamic>
      </restore-path>
    </cs-sr-te-policy>
  </config>
```

**Step 8** When you have finished editing the file and saved your changes, navigate to **Services & Traffic Engineering** > **Provisioning** > **SR-TE** > **Circuit-Style Policy** again.

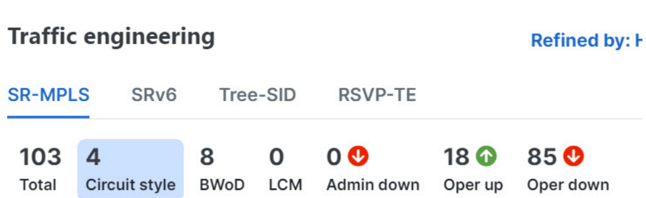
**Step 9** Click again. In the **File Name** field, enter the path and file name of your modified template file, or click **Browse** to locate and select it. Then click **Import**.

# Step 5: View Circuit Style SR-TE policies on the topology map

Next, use Crosswork to visualize the NCS1-NCS3 Circuit Style SR-TE policy and isolate it on the map. To make this step more realistic and demonstrate how to focus on just one policy, the scenario assumes that we have multiple active Circuit Style SR-TE policies, not just the policy we created. We'll also view the Circuit Style SR-TE policy details, including endpoints, bandwidth constraints, IGP metrics, and candidate (Active/Working and Protect) paths.

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

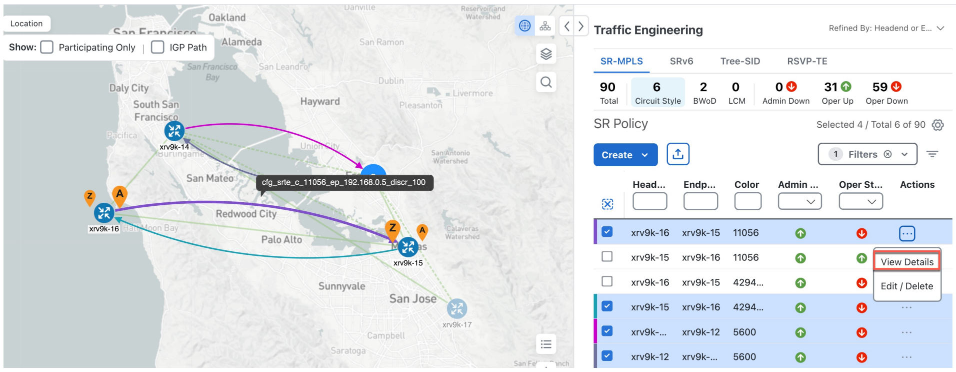
*Figure 96: View Circuit Style SR-TE policies*



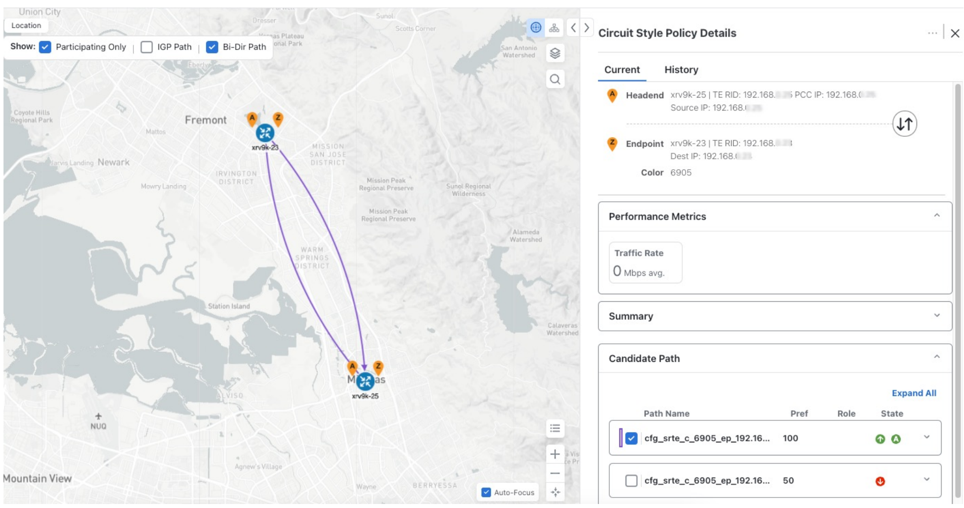

The policy table lists all of the Circuit Style SR-TE policies.

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

**Note** You cannot edit or remove Circuit Style SR-TE policy configurations that have been created directly on the device.



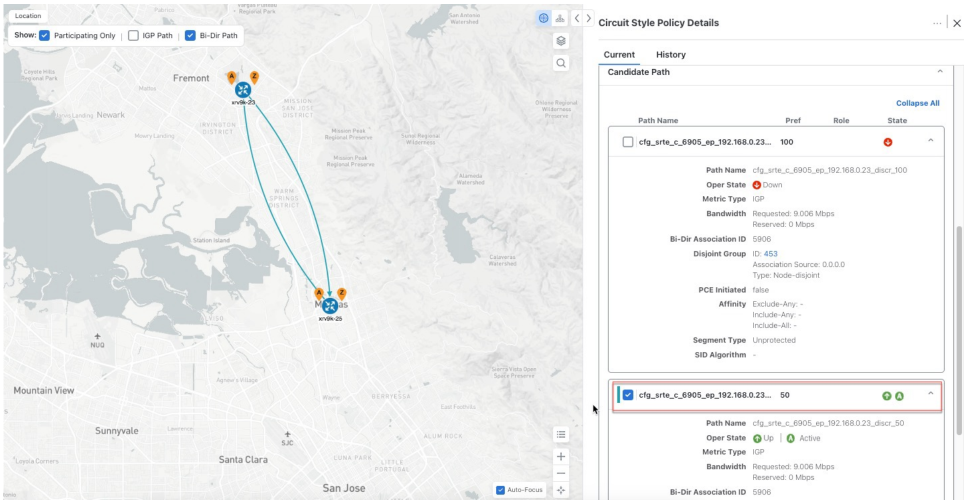

The **Circuit Style Policy Details** window is displayed in the side panel. By default, the Active path is displayed in the topology map and shows the bidirectional paths (Bi-Dir Path checkbox is checked) on the topology map. The Candidate Path list displays the Active (path that currently takes traffic) and Protected paths.




**Note** The Bandwidth Constraint value can differ from the bandwidth you requested if the value was increased and insufficient resources existed to satisfy demand on all Active and Protected candidate paths.

**Step 3** View Candidate path configuration details.

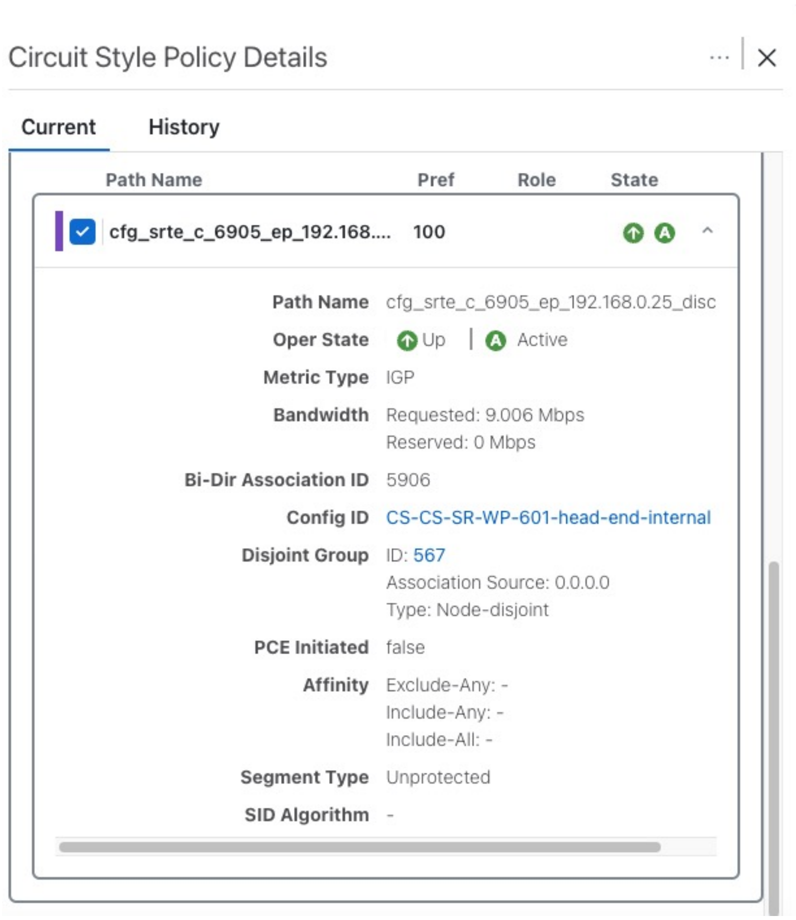
a) The **Circuit Style Policy Details** window allows you to drill down to view more information about the candidate paths. The operational (Oper State Up) candidate path with the highest preference will always be the Active path. In this example, the Protected path (with preference 50) is currently the Active path and is displayed on the topology map. Notice that it is designated with a green "A" icon under State to clearly indicate it is currently the operational Active path. Click **Expand All** to view more information about both paths.



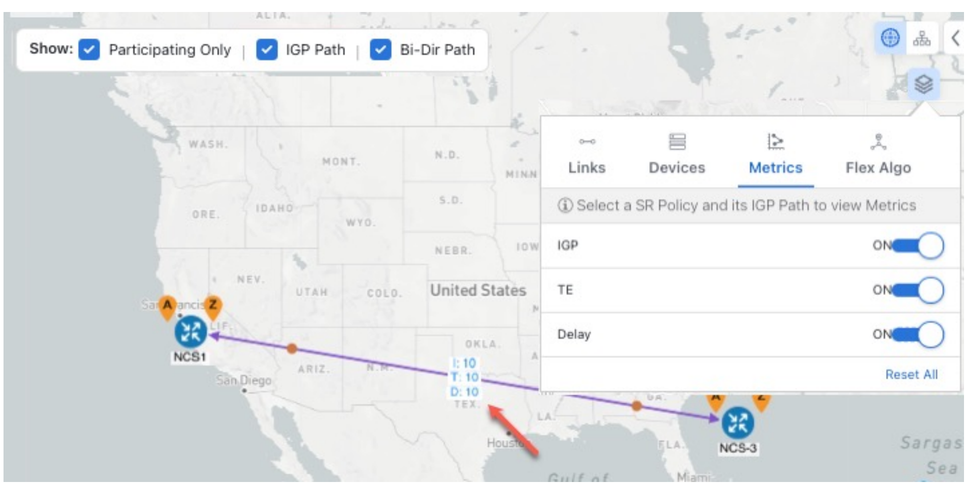

**Note**

- First preference paths are shown as purple links.
- Second preference paths are shown as blue links.
- Third preference paths are shown as pink links.

If the Circuit Style SR-TE policy configuration was done through the UI, you have the option to view the Circuit Style SR-TE policy configuration. To see the configuration, click the link next to **Config ID**. For example:




**Step 4** To view the physical path and metrics between endpoints of the selected policies, click to turn applicable metrics on and check the **IGP Path** checkbox.



# Step 6: Verify Circuit Style SR-TE policy bandwidth utilization

Let's verify that the reserved bandwidth pool settings we defined when enabling Circuit Style SR-TE (see Step 1: Enable Circuit Style Manager, on page 114) are configured properly. We can also check how much bandwidth is either in use or still available.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Traffic Engineering** > **SR-MPLS**. Then, under the **SR-MPLS** column, click **Circuit style**. The SR Policy table lists all CS SR policies.

**Step 2** In the SR Policy table, check the check box next to the participating device whose details you want to see.

**Step 3** On the topology map, click on a participating Circuit Style SR-TE policy node to display the device details for that node.

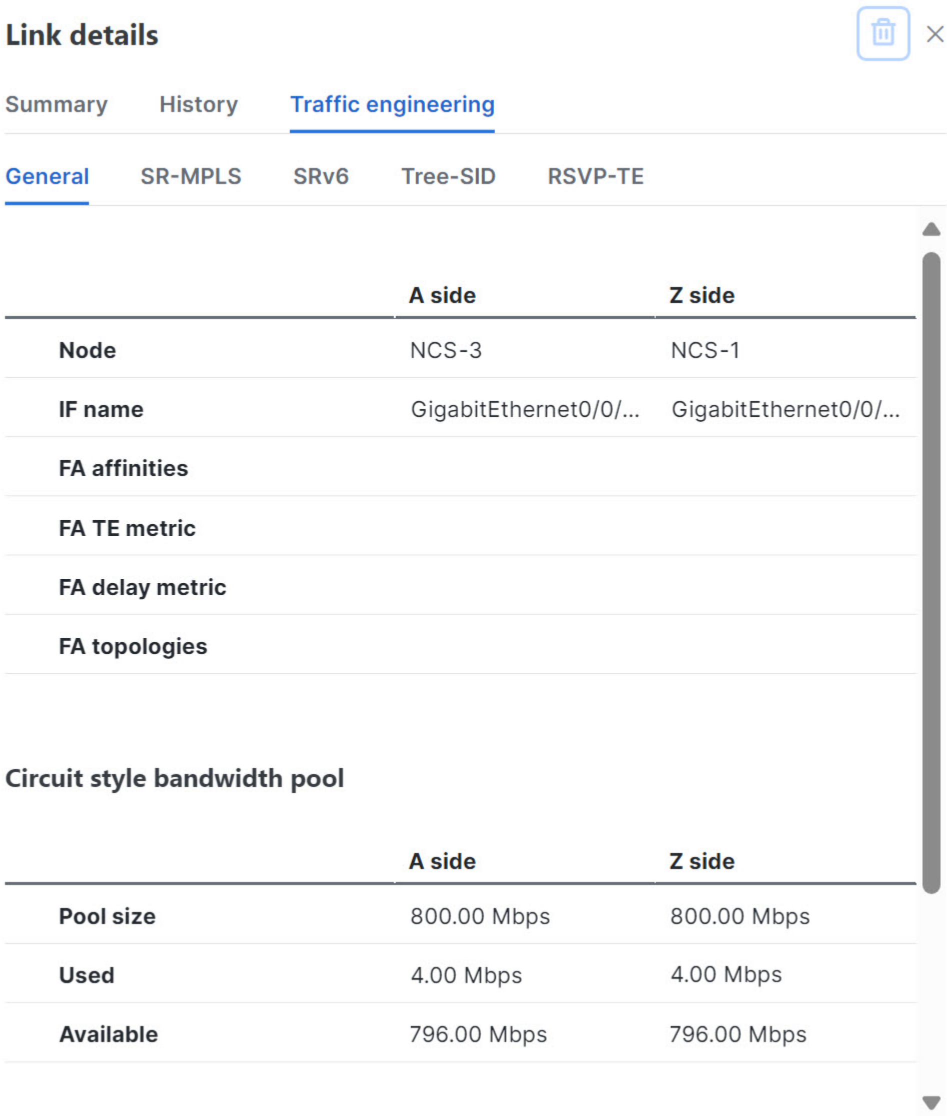
**Step 4** On the **Device details** page, click the **Links** tab to display the list of CS-SR and other links on the participating node. Then click on the link whose details you want to see. The **Link details** list displays a **Summary** of the link information.

**Step 5** On the **Link details** page, click on the **Traffic engineering** tab, then the **General** tab. The link details list displays detailed information for the link.

Under **Circuit style bandwidth pool,** you can see the reserved bandwidth pool size, the amount of bandwidth currently being used, and the amount of bandwidth (of the total allocated to Circuit Style SR-TE policies) that is still available.

This example shows the reserved bandwidth pool size as 800 Mbps for NCS-3 and NCS1. The configured settings were earlier defined as 80% for the bandwidth pool size. Since the interfaces on this circuit are 1 Gbps, we can confirm that Circuit Style SR-TE has correctly allocated 80 percent of the bandwidth for these two interfaces.

*Figure 97:*



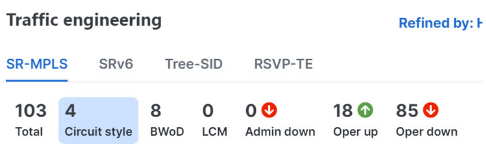

# Step 7: Trigger Circuit Style SR-TE path recomputation

Circuit-Style policies are static in nature, meaning once the paths are computed, Crosswork will not re-compute them automatically. Changes in your network topology or operational status may affect the previously computed Working and Protected paths to the extent that you want Crosswork to re-compute and optimize them for the new situation. In this step, we demonstrate how to re-optimize paths to accommodate these types of changes.

For more details on the logic CSM employs in these cases, see What happens when path failures occur?, on page 112.

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

*Figure 98: Select Circuit Style tab*




**Step 2** The SR Policy table displays the status of each of the Active CS-SR policies. One of them is Operationally down.

**Step 3** From the **Actions** column next to the CS-SR TE policies whose Operational State is **Down**, click ··· > **View details**.

Crosswork displays the **Circuit style policy details** window in the side panel. By default, the topology map shows the Active path and the bidirectional paths on the topology map (for these to appear, the **Bi-Dir path** checkbox in the topology map's **Show** panel must be checked). The **Candidate path** list at the bottom of the side panel displays the Active (Working) and Protected paths.

In the Summary panel, click the **See more** link to get a closer look at the type of Summary details available. The Candidate Path list displays the Active and Protected paths.

**Step 4** To have Crosswork re-optimize these paths: Click ··· at the top of the **Circuit style policy details** panel and select **Re-optimize**. Click **Yes** when prompted to confirm your selection.

# Summary and conclusion

In this scenario, we observed how to use Circuit Style Segment Routing policies to reserve bandwidth for high-priority services and traffic in the network. CS-SR removes the need to manually track and calculate high-priority traffic paths, but it still gives you control over how those paths are calculated and optimizes bandwidth usage on each path. You can use these policies to ensure that bandwidth is dedicated to these services. As traffic changes, Circuit Style policies warn you when your dedicated "circuit" paths fail and allow you to re-optimize them as needed.

CHAPTER 5

# Network Maintenance Window

This section explains the following topics:



## Overview

**Objective**

Schedule and automate maintenance workflows with minimal network interruption and most efficient results.

**Challenge**

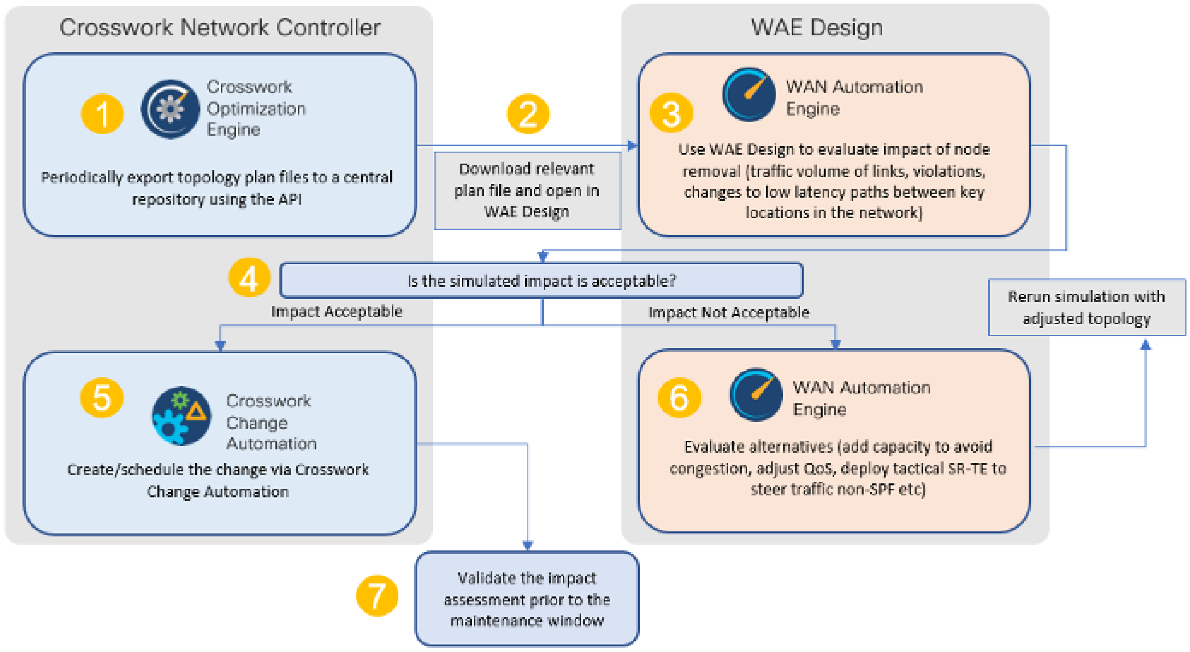
Maintenance activities typically require system downtime and temporary disruption of services. Keeping downtime and disruption to a minimum is critical but challenging. Therefore, maintenance activities occur during a carefully calculated optimal time slot, usually when activity is at its lowest.

**Solution**

Change Automation and Health Insights provide the functionality needed to automate the scheduling and execution of maintenance tasks (see Crosswork Network Controller 7.0 Closed-Loop Network Automation Guide for further information on Change Automation and Health Insights). Planning the optimal time for maintenance activities can be done successfully using Cisco WAE Design to simulate "what-if" scenarios based on timed topology snapshots exported from Crosswork Network Controller using APIs.

> **Note** Cisco WAN Automation Engine (WAE) is now also known as Crosswork Planning.

**How does it work?**




- Using Crosswork Network Controller APIs, you can create topology snapshots (plan files) that capture and represent the topology state at a given time, including the IGP topology and interface level statistics (traffic load). For impact analysis purposes, these snapshots should represent a time period to be evaluated for an upcoming maintenance activity. For example, if you are planning a router upgrade at midnight on a Monday, you would take snapshots from several Mondays at midnight to evaluate typical traffic loads. You can export these plan files to a central storage repository, where a library of topology plan files can be stored for a specified period.
- Cisco WAE Design allows you to explore "what-if" scenarios relevant to maintenance window planning. For example, if a router is upgraded, Cisco WAE Design can simulate the resulting traffic load on the remaining devices after traffic is diverted from the upgraded device. You can also explore the impact of deploying tactical traffic engineering policies to further optimize the topology during the maintenance window. For more information, contact your Cisco Customer Experience representative.

**Additional resources**

Cisco WAE Design documentation

Cisco Crosswork Network Automation API Documentation on Cisco Devnet

# Scenario: Install an SMU during a scheduled maintenance window

**Scenario context**

In this scenario, we will first use Cisco WAE Design to evaluate the impact of removing a Provider node from the network during a specific time frame to install a Cisco SMU (Software Maintenance Upgrade) on the device. We will choose a predefined Crosswork Playbook to automate the SMU installation on the device, and schedule it to run during the predetermined maintenance window.

### Assumptions and prerequisites

The high-level requirements that must be met to enable this scenario include:

- Change Automation is installed and running.
- You have access to Cisco WAE Design.
- The Device Override Credentials are set for Crosswork Network Change Automation, so that the Playbook can be run. From the main menu, choose **Administration** > **Settings** > **System Settings** > **Network Automation** to set these credentials.

# Step 1 Download topology plan files for impact analysis

When considering when to take down a device for maintenance so that there will be the least impact on the network, you need information about the traffic trends around that device at the targeted time. Using the Cisco Crosswork Optimization API, you can download plan files that capture a snapshot of the network topology at that time. If you download plan files at the same time over a period of time, you can use Cisco WAE Design to analyze the traffic trends. Based on this analysis, you can decide whether the impact on the network would be acceptable or not.

Refer to Cisco Crosswork Network Automation API Documentation on Cisco Devnet for more information about the API.

The input for this scenario is as follows:

---

**Step 1** Prepare the input required to download the plan file. You need to specify the version of Cisco WAE design that you will be using for analysis and the format in which you want the plan file, either `txt` or `pln`.

**Note** If you download the plan file as a `txt` file, you can view it in any text editor. If you download it as a `pln` file, you can open it only in Cisco WAE design.

The input for this scenario is as follows:

```
'{
  "input": {
    "version": "7.3.1",
    "format": "txt",

    }
  }
}'
```

**Step 2** Invoke the API on the Crosswork Network Controller server using the input prepared in the previous step. For example:

```
curl --location --request POST
'https://10.194.63.198:30603/crosswork/nbi/optima/v1/restconf/operations/cisco-crosswork-
optimization-engine-operations:get-plan \
--header 'Content-Type: application/yang-data+json' \
--header 'Accept: application/yang-data+json' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsImlzRnJvbU5ld0xvZ2luIjoidHJ1ZSIsInBvbGljeV9pZCI6ImFkb
WluIiwiYXV0aGVudGljYXRpb25EYXRlIjoiMjAyMS0wMy0yMlQxNjozODozNy43NDY2MTZaW0dNVF0iLCJzdWNjZXNzZnV
sQXV0aGVudGljYXRpb25IYW5kbGVycyI6IlF1ZXJ5RGF0YWJhc2VBdXRoZW50aWNhdGlvbkhhbmRsZXIiLCJpc3MiOiJod
HRwOlwvXC9sb2NhbGhvc3Q6NTQ4OVwvU1NPIiwibGFzdF9uYW1lIjoic21pdGgiLCJjcmVkZW50aWFsVHlwZSI6IlVzZXJ
uYW1lUGFzc3dvcmRDcmVkZW50aWFsIiwiYXVkIjoiaHR0cHM6XC9cLzEwLjE5NC42My4xOTg6MzA2MDNcL2FwcC1kYXNoY
m9hcmQiLCJhdXRoZW50aWNhdGlvbk1ldGhvZCI6IlF1ZXJ5RGF0YWJhc2VBdXRoZW50aWNhdGlvbkhhbmRsZXIiLCJsb25
nVGVybUF1dGhlbnRpY2F0aW9uUmVxdWVzdFRva2VuVXNlZCI6ImZhbHNlIiwiY2hhbmdlX3B3ZCI6ImZhbHNlIiwiZXhwI
joxNjE2NDU5OTIwLCJpYXQiOjE2MTY0MzExMjAsImZpcnN0X25hbWUiOiJqb2huIiwianRpIjoiU1QtODQtOFVlWXMybEt
```

```
3R2d1Z3RIYjl4MzVmTFlNTGVVRlp6OURyNGpoeFcxakhsV01VYXdXSWgxbUdTd01aRC1tOEk1S2Z0amI2ZmlWTUhlYnBYY
jBMMFZqRFc2WVppUFVUbHRpNFVpZnNUeG9aQ284WWpPWEc2VlFjS0Mwb29lWjJhc3BWanMzYnA3bHo5VkhySlBCTzl5TDN
GcFRIWXRPeWJtVi1jYXMtMSJ9.Vi4k0w8KsOv5M_O8zBqWochT3k9V9Pn2NjSn5ES9c5Pf-
4ds0o4kk6xuZx5_cggauiEICuUMnzmRzneST-oAuA' \
--data-raw '{
  "input": {
    "version": "7.3.1",
    "format": "txt",
    "
    }
  }
}
```

**Step 3** Note the plan file content in the API response. It is encoded for security purposes and must be decoded before you can view the content.

```
{
  "cisco-crosswork-optimization-engine-operations:output": {
    "status": "accepted",
    "plan-file content": "
PE5ldHdvcms+ClByb3BlcnR5CVZhbHVlClRpdGxlCQpWZXJzaW9uCTcuMy4xCgo8TmV0d29ya09wdGlvbnM+Ck9wdGlvbg
lWYWx1ZQpFbmZvcmNlQWRqU0lETG9jYWxpemF0aW9uCVRSVUUKCjxDaXJjdWl0cz4KTmFtZQl<<<>>>Ob2RlQQlJbnRlcm
ZhY2VBCU5vZGVCCUludGVyZmFjZUIJQ2FwYWNpdHkJRGVsYXkJRGlzdMJTmV0SW50U05NUF9FcnJvcglOZXRJbnRTb3VyY
2UJTmV0SW50UkUwQ1BVMW0JTmV0SW50UkUwQ1BVNWZpZXIJQWxnb3JpdGhtCVJmbGFnCU5mbGFnCVBmbGFnCUVmbGFnCVZ
mbGFnCUxmbGFnCg=="
  }
}
```

**Step 4** Use a script to decode the plan file or copy the plan file content into a decoder. After decoding the plan file, you can see the model content to be used in Cisco WAE Design. It includes a full topology snapshot, including the devices, interfaces, links, LSPs, traffic levels, and other information.

**Step 5** Open the plan file in Cisco WAE Design, simulate the device going down, and observe the impact on the network. Refer to the Cisco WAE Design documentation for more information.

**Step 6** Based on the analysis, decide an optimal time to execute the SMU.

---

# Step 2: Schedule the SMU installation playbook run

If the simulated impact is acceptable, you can create and schedule the change by running a playbook through Change Automation. For this scenario, we will run a predefined playbook to install a Software Maintenance Update (SMU) on devices tagged under a certain geographic location (NY).

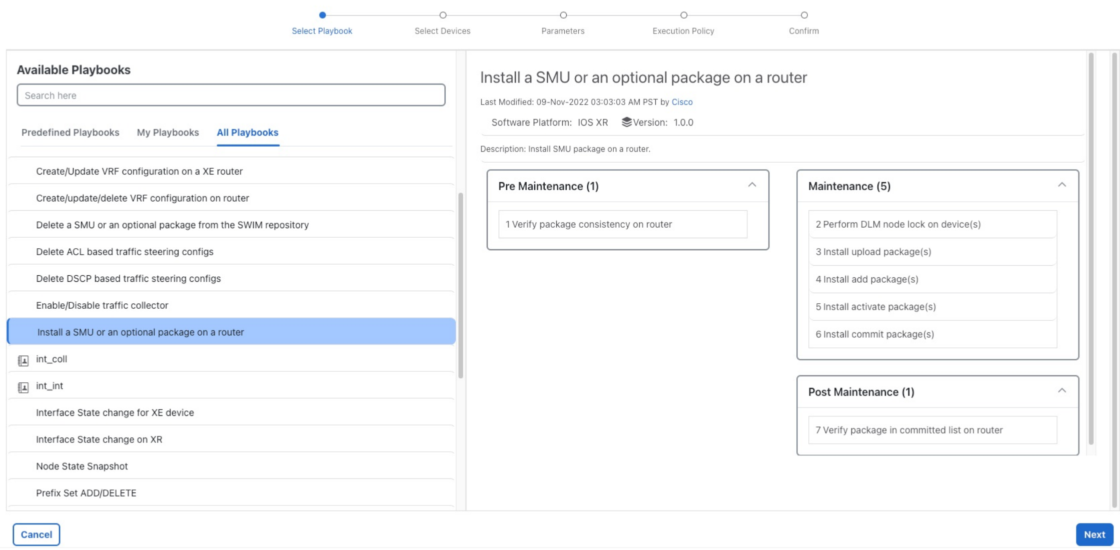
> **Note** If the predefined (stock) plays and playbooks do not meet your specific needs, you can create custom plays and playbooks. To create a custom play, from the main menu, choose **Network Automation** > **Play List**, and to create a custom playbook, choose **Network Automation** > **Playbook List**.

---

**Step 1** From the main menu, choose **Network Automation** > **Run Playbook**.

**Step 2** Browse the Available Playbooks list, and click the Install a SMU playbook. You can also filter using keywords to identify the playbook. Note that the playbook execution stages, supported software platform, software version, and individual play details are displayed on the right side.

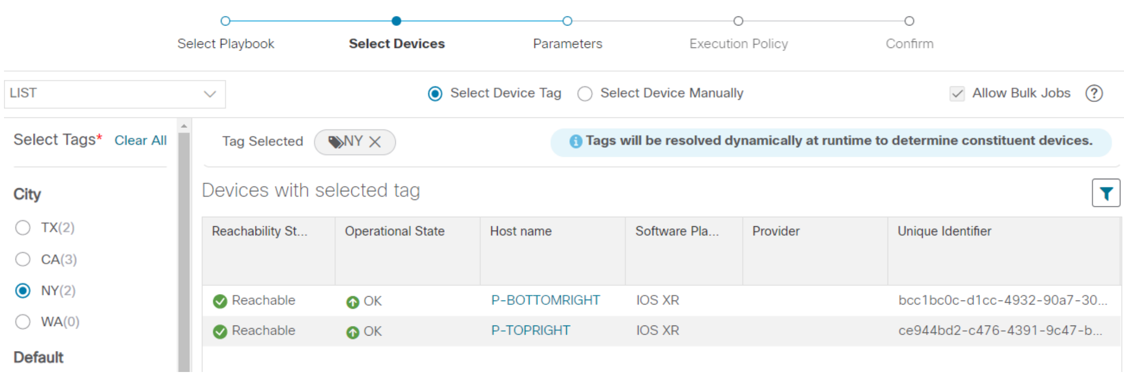
*Figure 99: Select playbook settings*




**Step 3** Click **Next** to go to the next task: Select Devices. All devices tagged with City: NY will be selected for SMU installation.

**Step 4** Under the City tag on the left, click **NY**. The devices tagged with NY are listed on the right and automatically selected.

*Figure 100: Select devices settings*



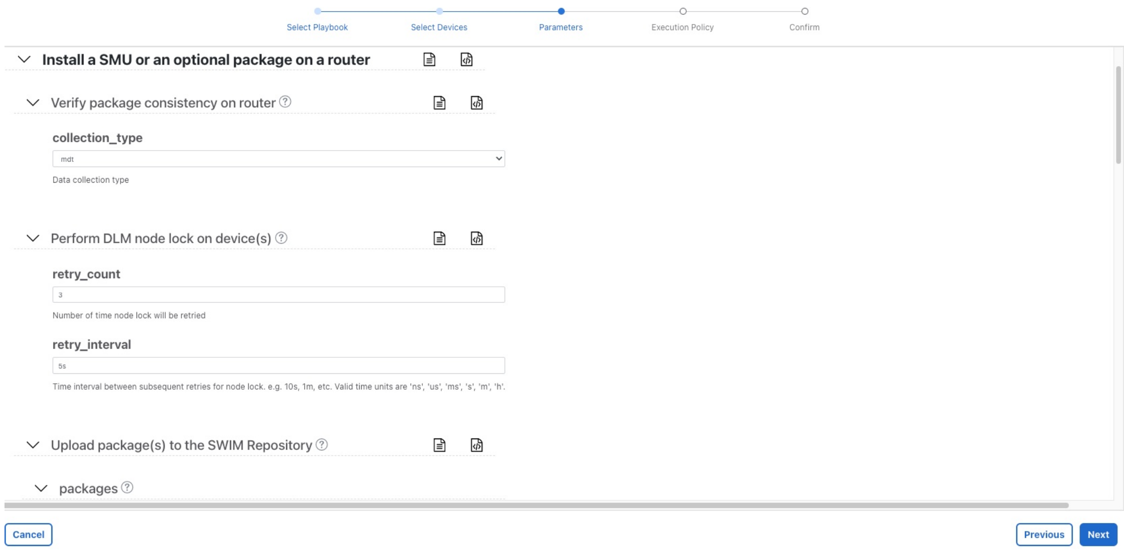

**Step 5** Click **Next** to go to the next task: **Parameters**.

**Step 6** Edit the runtime parameters to execute the SMU playbook. Alternatively, you can upload a JSON file that contains the parameter values. The following values are used specifically for this scenario. You can change them as required:

**a.** Under the **Install a SMU or an optional package on a router** play, set **collection_type** as **mdt**.

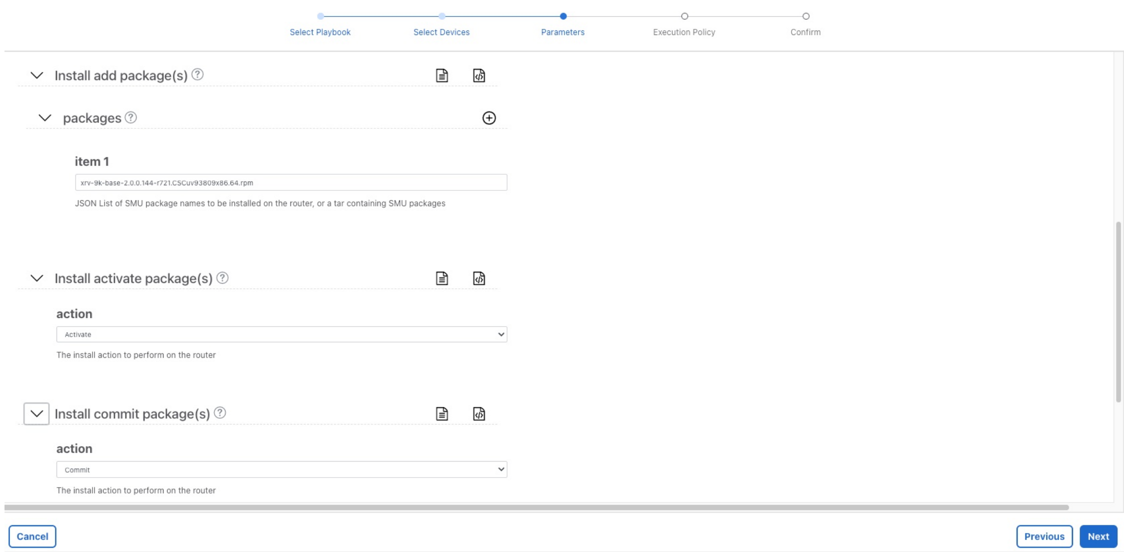
**b.** Under the **Perform DLM node lock on device(s)** play, set **retry_count** and **retry_interval** as **3** and **5s**, respectively.

*Figure 101: Parameters settings*




c. Under the **Install add package(s)** play, enter the SMU package name in **item 1**.

*Figure 102: Install add package(s)*




d. Under the **Install activate package(s)** play, click the piece of paper symbol, and set **action** to **Activate**.

e. Under the **Install commit package(s)** play, set the **action** to **Commit**.

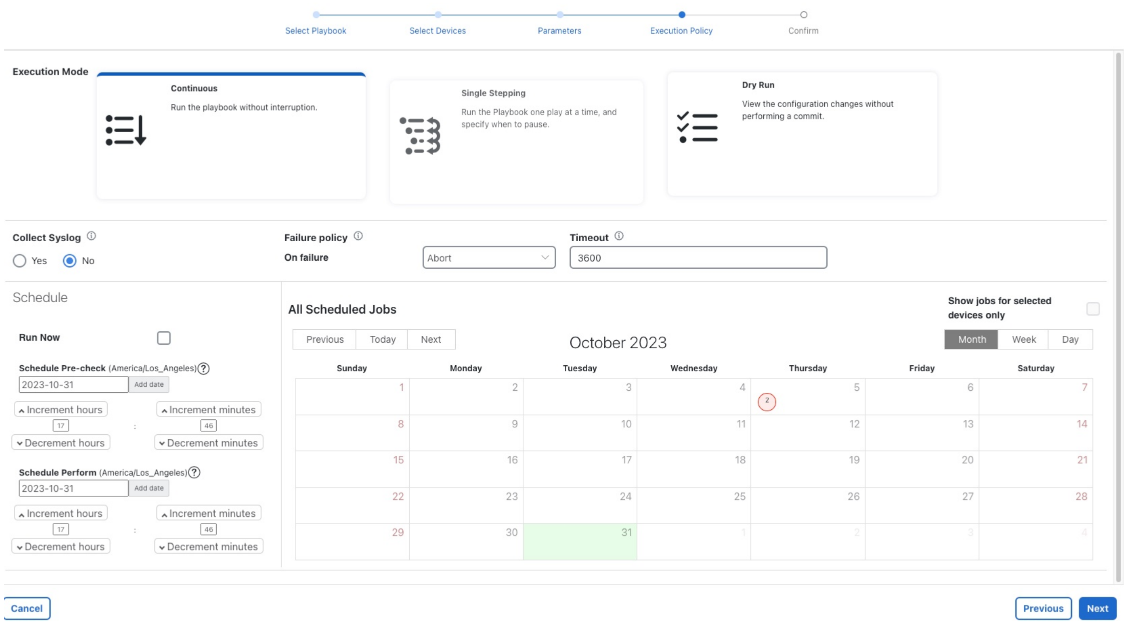
f. Under the **Verify package in committed list on router play**, set **collection_type** to **mdt**.

**Step 7** Click **Next** to go to the next task: **Execution Policy**.

**Step 8** Set the **Execution Mode** to **Continuous**. This will set the playbook to run uninterrupted, with no pauses. Under **Failure policy**, select the action you want taken if the execution fails: **Abort** or **Complete Roll Back**.

**Step 9** Under **Schedule**, set the playbook to execute for the optimal time calculated during the impact analysis stage. Uncheck the **Run Now** option. Note the calendar at the right and the timers that let you **Schedule Pre-check** and **Schedule Perform** play execution dates and times.

***Figure 103: Execution policy settings***

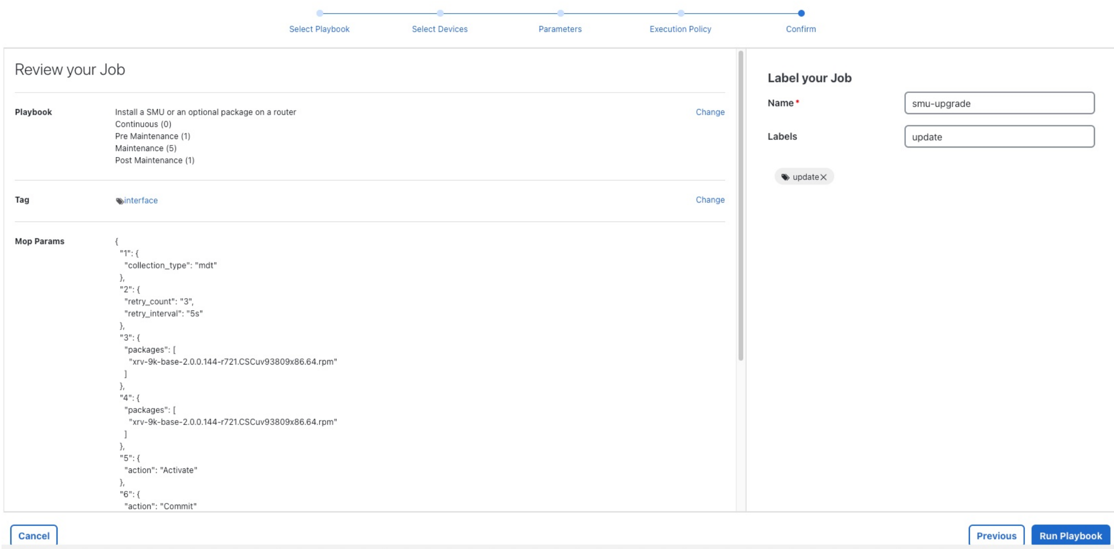



**Step 10** Click **Next** to go to the next task: Confirm Job.

**Step 11** Review your job details. Label your job with a unique name.

**Step 12** When you are finished, click **Run Playbook**. The SMU installation is now scheduled to run during the planned maintenance window.

*Figure 104: Confirm job*



# Step 3 Verify the SMU job status

**Step 1** After the scheduled maintenance window time, from the main menu, choose **Network Automation** > **Automation Job History**. Under **Job Sets**, check that the job status icon on the SMU install job is green, indicating that the scheduled job has run successfully.

**Step 2** Select the SMU install job. Note the Job Set details on the right side. Click the link under the job **Execution ID** for job details.

**Step 3** Double-check that the correct SMU has been installed by executing the `show install active summary` and `show install committed summary` commands on the device. If the installation was successful, the SMU will appear in the list of packages. The following figure shows example outputs from these commands:

*Figure 105: Example output*

```
RP/0/RP0/CPU0:CX-AA-PE4#show install active summary
Mon Apr 12 11:09:20.198 EDT
    Active Packages: 12
        ncs5500-xr-6.6.3 version=6.6.3 [Boot image]
        ncs5500-ospf-2.0.0.0-r663
        ncs5500-mpls-2.1.0.0-r663
        ncs5500-eigrp-1.0.0.0-r663
        ncs5500-isis-2.2.0.0-r663
        ncs5500-li-1.0.0.0-r663
        ncs5500-mpls-te-rsvp-4.1.0.0-r663
        ncs5500-mcast-3.1.0.0-r663
        ncs5500-mgbl-3.0.0.0-r663
        ncs5500-k9sec-3.1.0.0-r663
        ncs5500-routing-4.0.0.17-r663.CSCvr43225
        ncs5500-mpls-te-rsvp-4.1.0.17-r663.CSCvr43225

RP/0/RP0/CPU0:CX-AA-PE4#show install committed summary
Mon Apr 12 11:09:27.092 EDT
    Committed Packages: 12
        ncs5500-xr-6.6.3 version=6.6.3 [Boot image]
        ncs5500-ospf-2.0.0.0-r663
        ncs5500-mpls-2.1.0.0-r663
        ncs5500-eigrp-1.0.0.0-r663
        ncs5500-isis-2.2.0.0-r663
        ncs5500-li-1.0.0.0-r663
        ncs5500-mpls-te-rsvp-4.1.0.0-r663
        ncs5500-mcast-3.1.0.0-r663
        ncs5500-mgbl-3.0.0.0-r663
        ncs5500-k9sec-3.1.0.0-r663
        ncs5500-routing-4.0.0.17-r663.CSCvr43225
        ncs5500-mpls-te-rsvp-4.1.0.17-r663.CSCvr43225

RP/0/RP0/CPU0:CX-AA-PE4#
```

# Summary and conclusion

In this scenario, we saw how to plan for a maintenance window. Here, we took down a device for maintenance and scheduled an SMU installation. The goal is to cause as little impact to the network traffic as possible. To analyze the impact on the network, we showed how to download snapshots of the network topology (plan files) at the target time for the maintenance window. The plan files can then be analyzed using Cisco WAE design.

Assuming the impact was acceptable, we selected a predefined playbook to install the SMU on specific devices and scheduled it for the planned maintenance window.

CHAPTER 6

# Programmable Closed-Loop Remediation

This section explains the following topics:



## Overview

**Objective**

Detect anomalies and generate alerts that can be used for notifying an operator or triggering automation workflows.

**Challenge**

Discovering and repairing problems in the network usually involves manual network operator intervention and is time-consuming and error prone.

**Solution**

Incorporating Cisco Crosswork Change Automation and Cisco Crosswork Health Insights into Cisco Crosswork Network Controller gives service providers the ability to automate the process of discovering and remediating problems in the network by allowing an operator to match an alarm to pre-defined remediation tasks. These tasks will be performed after a defined Key Performance Indicator (KPI) threshold has been breached. Remediation can be implemented with or without the network operator's approval, depending on the setting and preferences of the operator.

Using such closed-loop remediation reduces the time taken to discover and repair a problem while minimizing the risk of making a mistake and creating an additional error through high-stakes manual network operator intervention.

**How Does it Work?**

**Smart Monitoring**

- The Smart Monitoring feature helps operators collect, filter, and present the data in useable formats, such as graphs and tables. Operators can remain focused on their business goals while the configuration required for the data collection is done by the Cisco Crosswork Network Controller and Cisco Crosswork Change Automation and Cisco Crosswork Health Insights using the feature Zero-touch telemetry.

- By using a common collector to collect network device data over SNMP, CLI, and model-driven telemetry, and making it available as modelled data described in YANG, duplicate data collection is avoided, optimizing the load on both the devices and the network.
- Recommendation Engine analyzes network device hardware and software, configuration, and employs a pre-trained model built from data mining, producing KPI relevant recommendations facilitating per use-case monitoring.
- KPIs cover a wide range of statistics from CPU, memory, disk, layer 1/2/3 network counters, to per protocol, LPTS and ASIC statistics.

**Smart Filtering**

- Cisco Crosswork Health Insights builds dynamic detection and analytics modules that allow operators to monitor and see alerts on network events based on user-defined logic (KPI).
- Key Performance Indicators (KPIs) Alerting Logic can be:
  - Simple static thresholds (TCA); e.g., CPU load above 90 percent.
  - Moving average, standard-deviation, and percentile based, etc., e.g., CPU load above mean and staying there for five minutes.
  - Streaming jobs which provide real-time alerts or batch jobs which run periodically.
  - Customized for threshold values and visualization dashboards.
  - Customized operator-created KPIs based on business logic.
  - TCAs can be exported or integrated with other systems via HTTP, Slack, and socket interfaces.
- KPIs are associated with dashboards, which provide real-time and historical views of the data and corresponding TCAs.
- KPIs also provide purpose-built dashboards that go beyond raw data and provide valuable information in various infographic style charts and graphs useful for triaging and root-causing complex issues.

**Smart Remediation**

- Health Insights KPIs can be associated with Cisco Crosswork Change Automation playbooks, which can be either executed manually or via auto-remediation. Remediation workflow could be used to fix the issue or collect more data from the network devices. By proactively remediating the situation, instead of resorting to ad hoc debugging and unscheduled downtime, operators can save time and money, providing better QOE to their customers.
- Health Insights does the correlation of alerts or anomalies on the topology of the network, allowing easy visualization of the impact of events.

# Scenario: Achieve predictive traffic load balancing using segment routing affinity

**Scenario context**

To maintain smooth and optimal traffic flow, operators need to be able to monitor traffic on the interfaces, identify errors such as CRC, watchdog, and overrun, and then reroute the traffic so that the SLA is maintained. This process can be automated using the Cisco Crosswork Network Controller with the installed Cisco Crosswork Health Insights and Cisco Crosswork Change Automation applications.

**Assumptions and prerequisites**

Cisco Crosswork Health Insights and Cisco Crosswork Change Automation must be installed and running.

# Workflow

The following is a high-level workflow for executing this scenario:

**Step 1** Deploy Day0 ODN templates on edge nodes with dynamic path calculation delegated to SR-PCE and the ODN template configured to exclude links that are tagged with a specific affinity, such as RED affinity. ODN allows a service head-end router to automatically instantiate an SR-TE policy to a BGP next-hop when required (on-demand). The ODN template defines the required SLA using a specific color.

For information on creating an ODN template, refer to Step 1 Create an ODN template to map color to an SLA objective and constraints, on page 30 in Scenario: Implement and maintain SLA for an L3VPN service for SR-MPLS (using ODN), on page 28.

**Step 2** Create an L3VPN route policy to specify the prefixes to which the SLA applies and mark them with the same color used in the ODN template. When traffic from the specified network with a matching color is received, paths are computed based on the SLA defined in the ODN template.

For information on creating a route policy, refer to Step 1 Create an ODN template to map color to an SLA objective and constraints, on page 30.

**Step 3** Provision an L3VPN across the required endpoints and create an association between the VPN and the route policy. This will connect the VPN to the ODN template that defines the SLA.

For information on provisioning an L3VPN, refer to Step 3 Create and provision the L3VPN service , on page 36.

**Step 4** Define and enable the KPIs on the devices. This will continuously monitor the uplink interfaces on the L3VPN endpoints.

For information on defining KPIs, see Crosswork Network Controller 7.0 Closed-Loop Network Automation.

**Step 5** When there is an error on monitored interfaces, mark the dirty link with RED affinity so that it will be excluded based on the specifications of the ODN template. This is achieved by creating a custom playbook. The Cisco Crosswork Network Controller identifies the interface where an error has occurred and generates an alert. This information is then used in a custom playbook to push the affinity configuration to the relevant router, creating a closed-loop automation process. This helps ensure that the customer does not experience any outages.

For information on defining playbooks, see the Crosswork Network Controller 7.0 Closed-Loop Network Automation.

**Step 6** Crosswork Network Controller will continue monitoring the link, and once there are no longer any alerts, the RED affinity tag can be removed. Another playbook should be defined for this purpose.

CHAPTER 7

# Automation of Onboarding and Provisioning of IOS-XR Devices Using ZTP

This section explains the following topics:



## Overview

**Objective**

Allow users to quickly, easily, and automatically onboard new devices and provision them using a Cisco-certified software image and a day-zero software configuration.

**Challenge**

Deploying and configuring network devices is a tedious task. It requires extensive hands-on provisioning and configuration by knowledgeable personnel, which is time-consuming, expensive, and error-prone.

**Solution**

Automate onboarding of new devices using Crosswork Zero Touch Provisioning (Cisco Crosswork ZTP). Cisco Crosswork ZTP allows users to provision networking devices remotely without a trained specialist on site. After establishing an entry for the device in the DHCP server and the ZTP application, all the operators must connect the device to the network, power it on, and press reset to activate the devices. A certified image and configuration are downloaded and automatically applied to the device. After it is provisioned in this way, the new device is onboarded to the Crosswork device inventory, which can be monitored and managed like other devices.

**How does it work?**

- Classic ZTP: The DHCP server verifies the device's identity based on its serial number and then offers to download the boot file and image. After the device is imaged, it downloads and executes the configuration file.
- Secure ZTP: The device and the Cisco Crosswork ZTP bootstrap server authenticate each other using the device's Secure Unique Device Identifier (SUDI) and Crosswork server certificates over TLS/HTTPS. After a secure HTTPS channel is established, the Crosswork bootstrap server allows the device to request to download and apply a set of signed image and configuration artifacts adhering to the RFC 8572 YANG

schema. After the image (if any) is downloaded and installed, and the device reloads with the new image, the device downloads configuration scripts and executes them.

- Plug and Play (PnP) ZTP: The Cisco PnP agent on the IOS-XE device and the Cisco Crosswork PnP Server authenticate each other over HTTP using a PnP profile supplied on a TFTP server. They then establish a secure connection over HTTPS, and the PnP agent downloads and installs images (optional) and configuration artifacts.

**Additional resources**

Detailed information is available in the ZTP chapter in Cisco Crosswork Network Controller 7.0 Device Lifecycle Management.

# Scenario: Use ZTP to onboard and provision new devices automatically

**Scenario context**

With the exponential growth of service provider networks and their rapid expansion into new customer sites and locations, there is a need to connect an ever-increasing number of edge devices. At the same time, functional sophistication is increasing, requiring more time to configure those devices and activate new services. Manual processes limit a service provider's ability to rapidly scale networks and roll out new services cost-efficiently.

In this scenario, we will onboard the new IOS-XR devices required to set up a new customer site in a remote location and go live without sending skilled technicians on time-consuming and costly on-site visits to complete the provisioning.

We will leverage the configuration of devices at existing customer sites that are already set up and operating to ensure that the Day0 configuration of the new devices includes whatever is necessary to get them up and running quickly and efficiently.

**Assumptions and prerequisites**

- Crosswork ZTP must be installed in your Cisco Crosswork Network Controller setup.
- For Classic ZTP, Crosswork and the devices must be deployed in a secure network domain. Secure ZTP does not have this requirement; it is secure across public networks.
- The Crosswork server must be reachable from the devices via an out-of-band management network or an in-band data network.
- If you also want to onboard devices to Cisco NSO, it must be configured as a Crosswork provider. When configuring the NSO provider, ensure that the provider property key is set to *forward* and the property value is *true*.

# ZTP Scenario: Workflow

This is a high-level workflow for onboarding IOS-XR devices using Cisco Crosswork Classic or Secure ZTP.

To onboard IOS-XE devices, or for more detailed information on these options, see the Zero Touch Provisioning chapter in the Cisco Crosswork Network Controller 7.0 Device Lifecycle Management guide.

**Step 1** Assemble and upload ZTP assets

a) Assemble the following assets before you begin:

- (Optional) Software images. For Classic ZTP, you can use Cisco IOS-XR versions 6.6.3, 7.0.1, 7.0.2, 7.0.12, and 7.3.1 or later. For Secure ZTP, use Cisco IOS-XR 7.3.1 or later (except 7.3.2 and 7.4.1).
- Configuration Files: SH, PY, or TXT files. You can specify up to three different configuration files for Secure ZTP.
- Credentials of the devices to be onboarded
- Serial numbers of the devices to be onboarded

For Secure ZTP only, also assemble:

- Owner certificates - Specifies your organization's CA-signed end-entity certificates installed on your devices, which binds a public key to your organization.
- Pinned domain certificate - Specifies your organization's CA- or self-signed domain certificate, with its public key pinned to your organization's DNS network domain. The PDC helps your devices verify that images and configurations downloaded and applied during ZTP processing come from within your organization.
- Ownership vouchers - Specifies nonceless audit vouchers that verify that devices being onboarded with ZTP are bootstrapping into a domain owned by your organization. Cisco supplies OVs when a request is submitted with your organization's PDC and device serial numbers.

b) To apply software images, upload the software images. From the main menu, choose **Device Management** > **Software Management** > **Images**.

c) To upload the configuration files, choose **Device Management** > **Zero Touch Provisioning** > **Configuration files**.

d) To upload device serial numbers, choose **Device Management** > **Zero Touch Provisioning** > **Serial Numbers & Ownership Vouchers** and click **Add serial number(s)**.

e) For Secure ZTP, upload your pinned domain certificate and owner certificates. From the main menu, choose **Administration** > **Certificate Management** and add your certificates.

f) For Secure ZTP, upload ownership vouchers. From the main menu, choose **Device Management** > **Zero Touch Provisioning** > **Serial Numbers & Ownership Vouchers** and click **Add voucher(s)**.

**Step 2** Create a ZTP profile combining an image file and configuration file

Crosswork uses ZTP profiles to automate imaging and configuration processes. While optional, creating ZTP profiles is recommended as the best way to combine a single image and configuration file based on a product or device family, such as the Cisco ASR 9000 or Cisco NCS5500. We recommend that you create only one day-zero ZTP profile for each device family, use case or role the devices serve in the network.

To create ZTP profiles, go to **Device Management** > **Zero Touch Provisioning** > **ZTP profiles**.

**Step 3** Prepare ZTP device entries for the devices to be onboarded

Depending on how many devices you are onboarding, you can prepare and import a CSV file or create device entries individually.

**a.** Go to **Device Management** > **Zero Touch Provisioning** > **Devices**.

**b.** Click **Total ZTP devices**. Then:

- To create a device entry file for many devices, click the **Import** icon and download the CSV template. Edit the template and add entries for each device you want to onboard. See the Zero Touch Provisioning chapter in the Cisco Crosswork Network Controller 7.0 Device Lifecycle Management guide for details on the file entries. Then click the **Import** icon again to import your device entry file.
- To create device entries one at a time, click the **Add devices** icon.

**Step 4** Set up DHCP for Crosswork ZTP

Before triggering ZTP processing, you must update your organization's DHCP server configuration file with the IDs for your ZTP device entries and the paths to the image and configuration files stored in the ZTP repository. This allows Crosswork and DHCP to identify these ZTP devices, respond correctly to each device's network connection requests, and download image and configuration files. For sample DHCP entries, see the Zero Touch Provisioning chapter in the Cisco Crosswork Network Controller 7.0 Device Lifecycle Management guide.

**Step 5** Initiate ZTP processing to onboard the devices

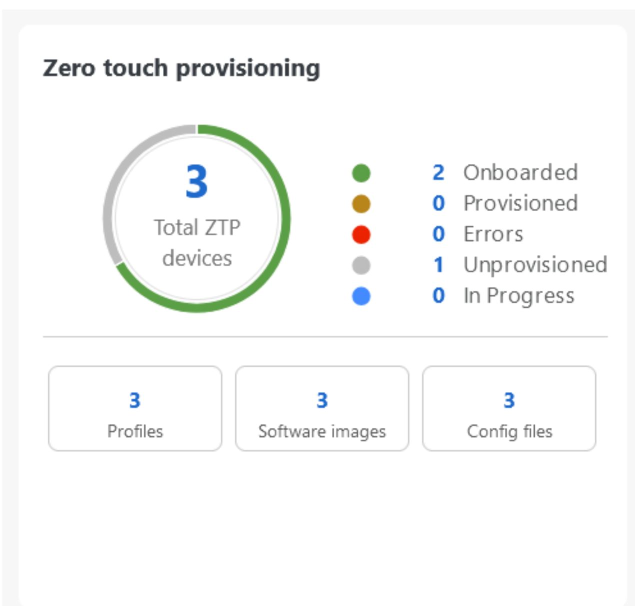
Initiate ZTP processing by rebooting each device to be provisioned: Power-cycle it or press the chassis reset button.

**Step 6** Monitor the ZTP processing status

You can monitor the progress of ZTP processing in the dashboard.

**a.** Click **Dashboard** in the main menu and look at the Zero Touch Provisioning dashlet.

*Figure 106: Zero Touch Provisioning*




**b.** Click on the Total ZTP devices number link to view the detailed status of the devices.

**Step 7** Verify your onboarded devices

From the main menu, choose **Device Management** > **Zero Touch Provisioning** > **Devices**. Click **Total ZTP devices**. All of your onboard devices should be listed.

You may need to edit the information for some devices. Some of the information needed for a complete device record either is not needed to onboard the device or is not directly available through automation. For example, geographical location data is defined using a set of GPS coordinates.

ZTP devices, after being onboarded, are automatically part of the shared Crosswork device inventory. You can edit them like any other device.

CHAPTER 8

# Visualization of Native SR Path

This section explains the following topics:



## Overview

### Objective

Visualize the actual path traffic flows physically through the topology map, even if traffic is on a native SR IGP path (not SR-policy) over inter-AS option C.

### Challenge

Visualizing the native SR IGP path is often an operational challenge. Without access to a streamlined and simple-to-use interface, diagnosing and troubleshooting the native path requires repeatedly logging in to network devices without a solution to improve efficiency.

### Solution

With the Path Query option, the objective is to visualize the native path using the traceroute SR-MPLS multipath command to get the actual paths between the source and the destination. With Cisco Crosswork Network Controller, a traceroute command runs on the source device for the destination TE-router ID and helps in retrieving the paths. By using native gRPC calls from the Crosswork server, you are able to get the paths from the device which assist in visualizing the native path through which the traffic flows. Since the traceroute command results in an operation that might take time to converge, Cisco Crosswork Network Controller provides an asynchronous user experience where you can send a request for such an operation and then be notified when the output is ready for inspection.

### How does it work?

- Create a new path query, defining the headend and endpoint devices to find the available Native SR IGP paths.
- Visualize the available Native SR IGP paths as defined by the query on the topology map.
- Inspect the available paths and review the Output, Nexthop, Source, Destination, and Hop Index information.

- Create additional path queries based on service type and instance and visualize the paths on the topology map.
- Troubleshoot any failed path queries.

# Scenario: Troubleshoot native SR IGP paths over Inter-AS Option C

**Scenario context**

Visualization of traffic flow paths is only readily available with manual tasks from different sources. Once traffic flow paths are attained, the data is often out of date. Crosswork Network Controller supports the creation of Path Queries, which you define within the Crosswork GUI. This allows visualization of actual SR IGP paths between the source and destination on a topology map. Crosswork Network Controller provides an asynchronous user experience, where the user is notified when results are ready for inspection. This facilitates rapid troubleshooting for issues with native traffic flows.

**Assumptions and prerequisites**

- The device should have IOS XR version 7.3.2.
- The device should have gRPC (Remote Procedure Call) enabled. To check, run “show grpc” on the device and follow these steps:
    - For gRPC without a secure connection: If gRPC is showing as not enabled, enable gRPC using the following commands: configure terminal; grpc; no-tls.
    - For gRPC with a secure connection: Upload security certificates to Crosswork Network Controller and connect to the device using the following commands: configure terminal; grpc.
- Your Crosswork Optimization Engine server should have the devices imported with gNMI (Network Management Interface) capability and gNMI connectivity.
    - Make sure the credential profiles include connectivity information for gNMI. From the main menu, choose **Device Management > Credential Profiles**. The Credential Profiles screen appears. Select a profile to edit. On the Edit Profile Devices screen, click **+ Add another**. For Connectivity type, select **GNMI**. Add the User Name, Password, and Confirm password information. Click **Save**.
    - Devices should have gNMI capability enabled in Crosswork Network Controller while attaching the device. From the main menu, choose **Device Management > Network Devices**. Select the device to edit. The Edit Device screen appears. From the required Capability list, select **GNMI**. Click **Save**.
    - Devices should have the gNMI connectivity information enabled. From the main menu, choose **Device Management > Network Devices**. Select the device to edit. On the Edit Device screen, under Connectivity details, click **+ Add Another**. For Protocol, select **GNMI** and add the IP Address / Subnet Mask information. Type the Port information and for Encoding Type, select **JSON**. Click **Save**.

# Workflow: Native SR IGP paths troubleshooting

**Step 1** Select **Services & Traffic Engineering** > **Path Query**. The **Path Query** dashboard appears.

**Step 2** Click **+ New query**. The **New Path Query** window appears, with query parameters displayed on the right and the topology map, with the mapped Device Groups, on the left.

**Step 3** Enter the device information in the required fields to find available native SR IGP service paths:

a) Select the **Headend** device from the list. For this example, select **P-Edge-A1**

b) Select the **Endpoint** device from the list. For this example, select **P-Edge-B2**

**Step 4** Click **Get paths**. The **Running Query ID** pop-up appears, indicating that Crosswork is processing your query.

Path queries can take time to complete. While your query is running, if you want to review past queries, select **View past queries**. Crosswork redisplays the **Path Query** dashboard. If you already had path queries in the list, you can view them while your new query runs in the background (indicated by the blue Running icon in the **Query State** column). When the new query's state turns green, click on the entry for it in the dashboard to view your results

I

**Step 5** When the pop-up changes to show that your query was successful, click **View results**. The **Path Details** panel appears, with corresponding **Available paths** details displayed below your **Headend** and **Endpoint** selections. Each found path will appear in its section in the **Available paths** portion of the panel. Meanwhile, the topology map on the left is filtered to display only the nodes and paths participating in the **Available paths**.

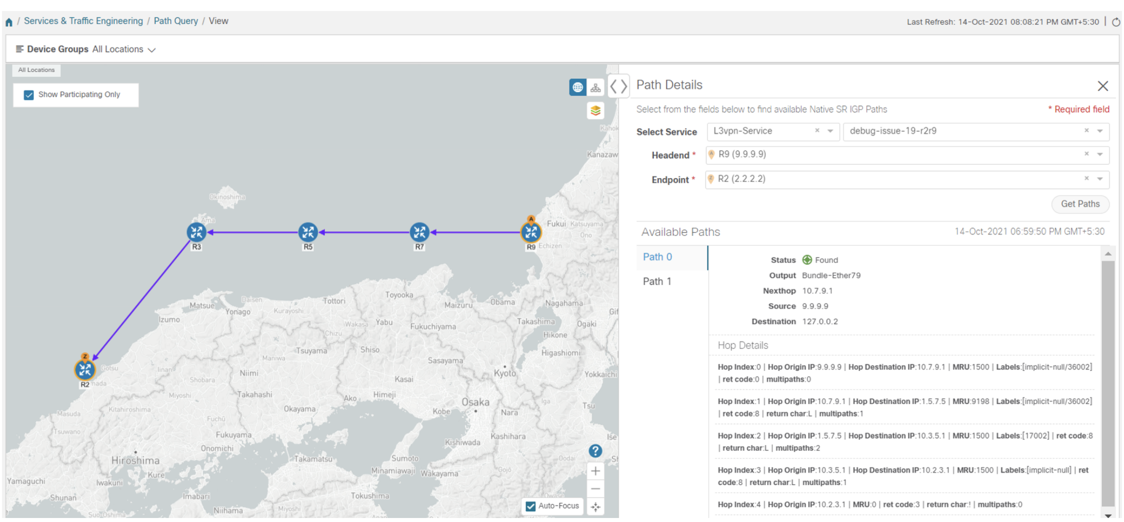
**Step 6** Click on the **Available paths** option whose details you want to review (for example, click **Path 0** or **Path 1**). Crosswork Status details for Output, Nexthop, Source, Destination, and Hop index information. When you select one of the available paths, the map will update with the corresponding Device Groups topology mapping of Path 0 and Path 1.

**Note** Ensure that the **Show participating only** check box is selected in the top-right corner of the map.

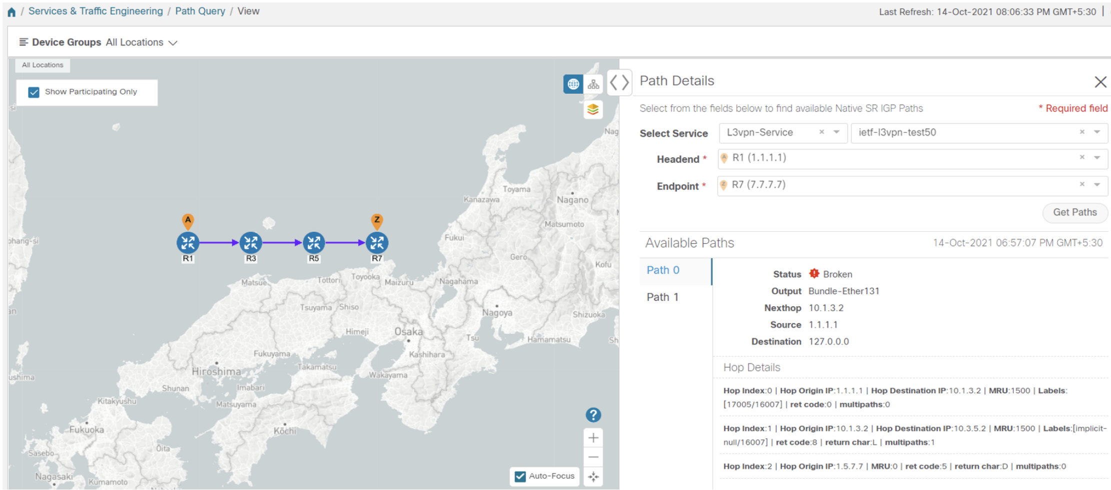
There are three likely status outcomes to a path query:

**Note** The screen captures below are independent examples not directly associated with the scenario's workflow.

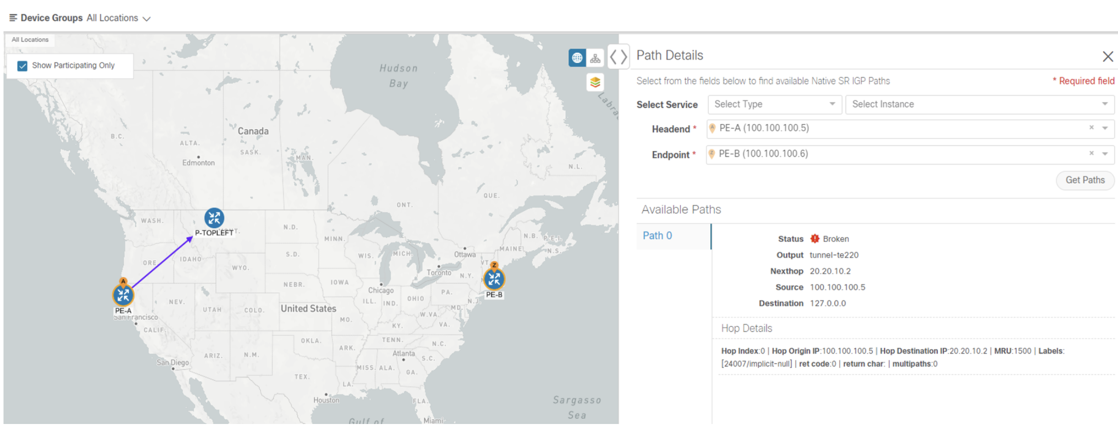
**a. Non-broken path (path is complete):** The Path Status shows as **Found**, with path hop details and an overlay shown.

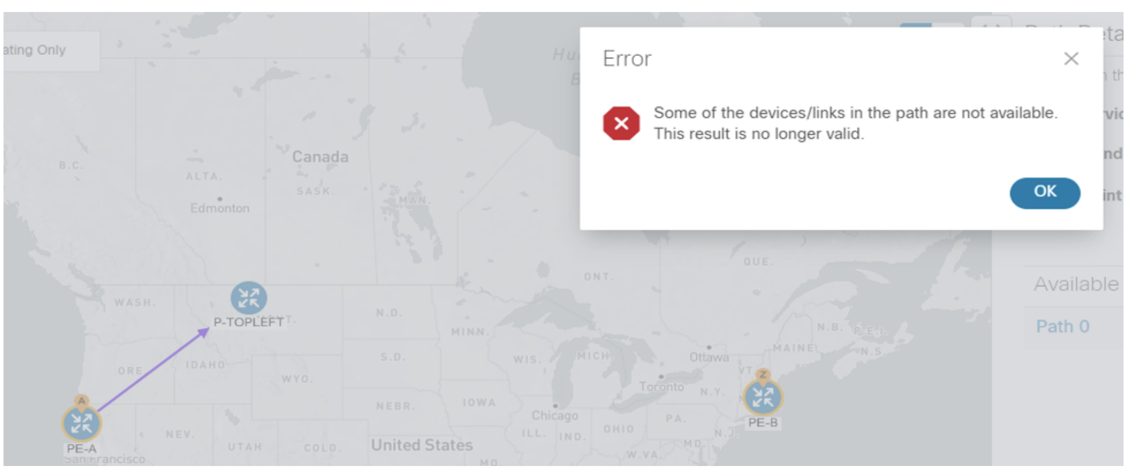


b. **Broken path (path is complete):** The Path Status shows as Broken, with path hop details and an overlay shown.




c. **Broken path (path is not complete):** The Path Status shows as Broken, with path hop details partially shown (depending on gNMI output for traceroute – see Step 17 for troubleshooting details) and overlay details partially shown. An error message will appear indicating that the devices and links are unavailable.




**Step 7** Select **Services & Traffic Engineering** > **Path Query** to return to the **Path Query** dashboard.

**Step 8** Ensure that the new path Query State column shows as completed with a green icon. The new path in the table will also show a Query ID link, both the corresponding Headend and destination Endpoint, and the Available Paths column will show 2 for both paths.

If a query state is broken, see the last step in the workflow for troubleshooting details.

**Step 9** As needed, click on the **Query ID** link or click and select **View Details** to again review the Path Details panel and map.

**Step 10** Create additional path queries by selecting **Services & Traffic Engineering > Path Query**. The Path Query Dashboard appears where the previous path queries are listed by Query ID.

**Note** On the Dashboard, set the **Automatically delete queries < X > hours after creation** option, where X is the number of hours. The maximum number of hours you can specify is **24**.

**Step 11** Click **+ New query**. The **New path query** panel appears on the right, and the mapped Device Groups panel appears on the left.

**Step 12** For **Select service**, select the type from the list. In this example, select **L2VPN-SERVICE**.

By utilizing **Select service**, the options are conveniently identified according to the relevant VPN service type when you later select the Headend and Endpoint.

**Step 13** For **Select service**, select the instance from the list. In this example, select **L2VPN_NM_P2P-NATIVE-210**.

The topology map will update to show the path between both servers. In this example, **P-Edge-B2** and **P-Edge-C3** are isolated on the map, showing the logical path.

**Step 14** Select the following from the list:

**a.** Headend: **P-Edge-B2**.

**b.** Endpoint: **P-Edge-C3**.

**Step 15** Click **Get paths**.

The Running Query ID pop-up appears.

**Step 16** Click **View results** when it becomes available. The Path Details panel appears with the corresponding Available Paths details, while the defined topology map appears with the available Native SR IG Paths on the left. This view shows the actual physical hops between B2 and C3 carrying the traffic.

**Step 17** To troubleshoot any Failed path queries appearing in the Path Query Dashboard's Query State column, select the "I" icon for error details.

In this example, the gNMI protocol is missing from the Connectivity Details for a previous path query with the Headend P-BOTTOMLEFT device and the Endpoint P-BOTTOMRIGHT devices. To troubleshoot the failed path query, do the following:

**a.** From the main menu, choose **Device Management > Network Devices**.

**b.** Find the device by Host Name and select the check box.

**c.** Click the Edit icon at the top of the table. The Edit Device window appears.

**d.** In this example, the Connectivity Details for Protocol is missing gNMI. Click **+ Add another** and type GNMI until it appears in the list. Select it.

**e.** Enter the IP Address / Subnet Mask information and Port field information.

f. Enter the Timeout field as **30**.

g. In the Endcoding Type list, type **JSON** until it appears in the list. Select it and click **Save**.

h. From the main menu, choose **Services & Traffic Engineering > Path Query**. The Path Query Dashboard appears.

i. Click **New query**. The New path query panel appears.

j. Select the following from the list:

   1. Headend device: **P-BOTTOMLEFT**.

   2. Endpoint device: **P-BOTTOMRIGHT**.

k. Click **Get paths.** The Running Query ID pop-up appears.

l. Click **View results** when it becomes available. The Path details panel appears with corresponding Available paths. The topology map now shows the available Native SR IG Paths on the left and is in completed state.

CHAPTER 9

# Provision, Visualize, and Analyze Tree Segment Identifier Policies in Multipath Networks

This section explains the following topics:



## Overview

Allow users to provision and visualize Tree Segment Identifier (Tree-SID) Segment Routing policies easily and quickly before associating the policies with an L3VPN service model.

**Objective**

To provision, visualize, and update static Tree-SID policies within your network using Crosswork Network Controller and associate the (mVPN) policies with an L3VPN service model. Provisioning Tree-SID policies through the Crosswork Network Controller UI allows for visualizing and analyzing multicast paths, root and leaf nodes, transit nodes, and link information among the nodes. This process provides a comprehensive view for creating, visualizing, updating, and maintaining point-to-multipoint (P2MP) network configurations. These static Tree-SID policies can be associated with a L3VPN service model and can be visualized and edited as needed using the Crosswork Network Controller UI.

**Challenge**

Keeping track of SR PCE and PE paths within networks is a challenge for video broadcasting and streaming service providers, who must use multipath protocols to replicate traffic and send it to different points in the network. To ensure a high-quality service, providers need to use difficult manual approaches to visualize, update, and maintain their point-to-multipoint (P2MP) network configurations. This approach slows response to network problems and increases costs.

**Solution**

Tree-SID is a method of implementing tree-like multicast flows over a segmented routing network. Using Tree-SID, an SDN controller (a device running SR-PCE using PCEP) calculates the tree. Each node (device) in the tree has a specific role in routing the multicast data through the tree. These roles include Ingress for the root or headend node, Transit or Bud for midpoint nodes that are not leaf nodes, and Egress for destination leaf nodes. The tree itself is assigned a single SID label representing all the tree segments and devices. The

SDN controller is highly flexible, as it understands the segmentation and can construct routing paths using any constraints that network architects can specify.

The most interesting use case for constraint-based Tree-SID use is where routers are configured to deliver two P2MP streams with the same content over different paths. Here, the multicast flow is forwarded twice through the network, each copy following a unique path. The two copies never use the same node or link to reach the destination, reducing packet loss due to network failures on any one of the paths.

Using Crosswork Network Controller, you can now create static tree-SID policies using the UI, associate Static mVPN tree-SID policies with a provisioned L3VPN service, and visualize, analyze, edit or delete your tree-SID policies to manage your multicast network actively.

**Note** **Static** and **Dynamic** mVPN Tree-SID policies can be associated with an L3VPN service model. In this workflow tutorial, only a Static mVPN Tree-SID policy will be associated, visualized, and analyzed with an L3VPN service model.

**How does it work?**

- Create a static Tree-SID policy using Crosswork Network Controller UI
- Visualize and validate the new static Tree-SID policy
- Associate your static mVPN Tree-SID policy with an L3VPN service model (or import an existing static or dynamic Tree-SID policy)
- Visualize and analyze the performance details of your static mVPN Tree-SID paths and nodes within the L3VPN service model
- Edit your existing static mVPN Tree-SID policy to enhance performance or correct issues with your Tree-SID L3VPN service model

# Scenario: Provisioning, visualizing, and analyzing Tree-SID policies in a point-to-multipoint L3VPN service

**Scenario context**

Without Crosswork Network Controller, provisioning and visualizing Tree Segment Identifier (Tree-SID) point-to-multipoint traffic flows is possible only using manual tasks from different sources. When manual tasks are restricted, creating Tree-SID policies, associating a policy with an L3VPN service model, and visualizing and editing the policy and service becomes more difficult. By using Crosswork Network Controller, you can bypass the time-consuming manual setup process and see the traffic flow paths without relying on outdated data from manual configurations. The Crosswork Network Controller can create and discover Tree-SID segmentation directly from network configurations and display the data flow map immediately. This makes it easier to troubleshoot issues with Tree-SID traffic flows quickly.

Crosswork Network Controller’s topology services use PCE topology and LSP data to discover and visualize pre-configured Tree-SID policies in your network. The PCE controller manages the layer 3 topology, LSP, and Tree-SID data using the BGP link state and supports initial discovery and notifications for the Tree-SID trees. Static Tree-SID policies can also be configured and associated with newly created or previously configured

L3VPN services directly in Crosswork Network Controller's UI. Likewise, based on the health of the service and policies, editing capabilities are also performed using the UI to troubleshoot and optimize model operations.

**Assumptions and prerequisites**

If your network has PCE and Tree-SID policies already configured on your devices, this workflow assumes, at a minimum, the following basic configuration options:

1. On all nodes involved in the Tree-SID path, irrespective of role:
    a. Enable Path Computation Element Protocol (PCEP)
    b. Enable Computation Client (PCC)
2. Under SR-PCE, on endpoints: Configure a P2MP SR static or dynamic Policy.
3. On all root and leaf nodes:
    - Enable multicast routing
    - Configure `interface vrf <vrf-number>`
    - Configure `router bgp` on topo nodes and PCE. On corresponding neighbors between PCE and PCC nodes, mention the configured `interface vrf <vrf-number>`.
    - Configure `route-policy <vrf-number>` and `PASS_ALL`
    - Under segment routing traffic engineering: Configure `ODN color <same as vrf-number>`
4. On all leaf nodes only: Configure router PIM, route-policy TREESID_CORE.

# Step 1 Create a static Tree-SID policy

If you are using preconfigured Static or Dynamic Tree-SID policies already configured on your devices, skip to Step 2 in the workflow. If you are configuring Tree-SID policies using the Crosswork Network Controller's UI, this task first creates a Static Tree-SID policy, each representing a leaf or root node, before you have the option to associate the policies with a L3VPN service model that can be visualized and edited as necessary:

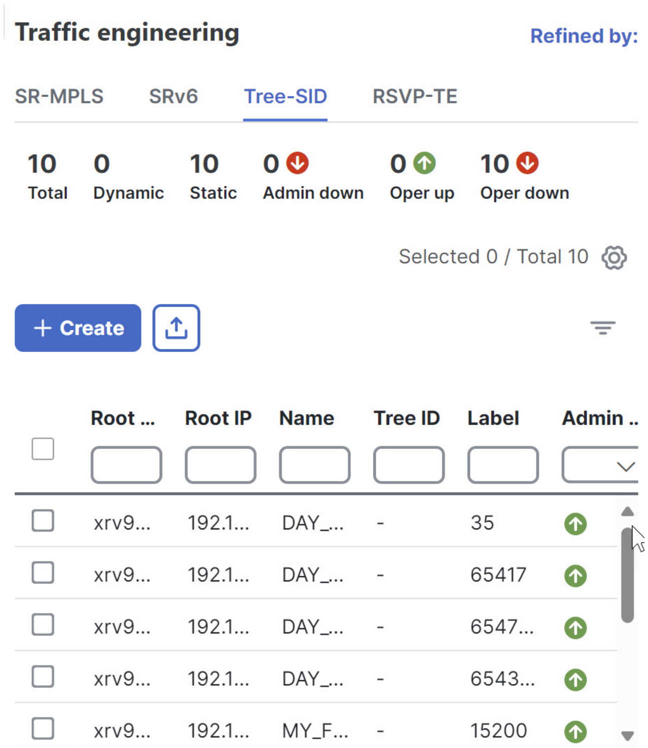
**Step 1** From the main menu, choose **Services & Traffic Engineering > Traffic Engineering**.

The logical map opens and the Traffic Engineering panel is displayed to the right of the map.

**Step 2** In the Traffic Engineering panel, select the **Tree-SID** tab.

The Traffic Engineering Tree-SID Policy screen appears.

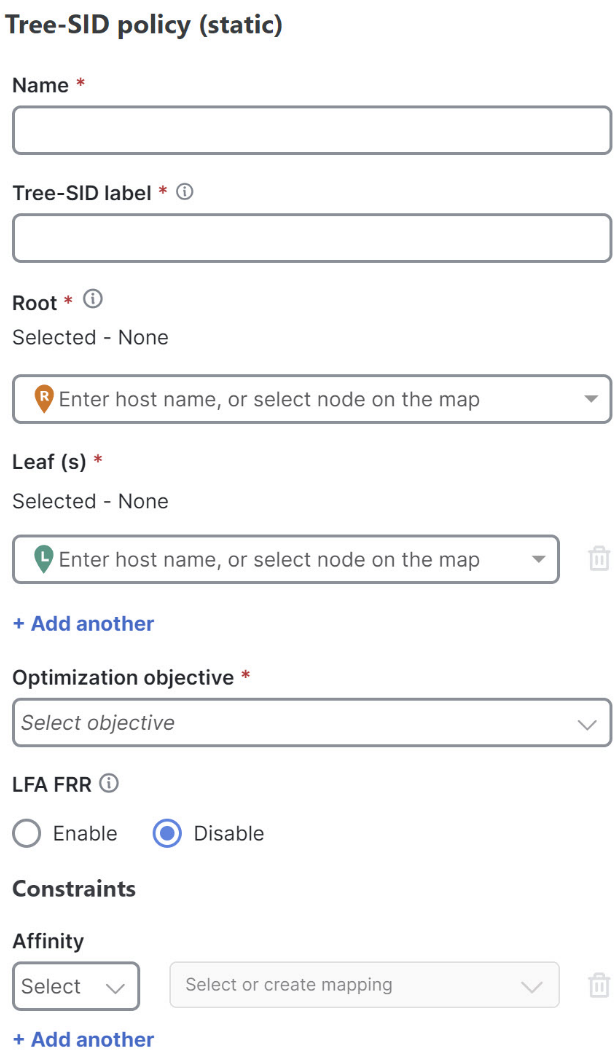
*Figure 107: Tree-SID policy*




**Step 3** Click + **Create**.

The New Tree-SID Policy (Static) screen appears.

*Figure 108: New Tree-SID policy (static)*




**Step 4** To enter or select the required Static Tree-SID policy values, do the following:

a) After providing a name for your new Static Tree-SID policy, in the Tree-SID Label field, assign the MPLS label associated with the Tree-SID policy (for example, **152001**).

   The Tree-SID Label must be in the range from 16 to 1048575.

b) In the Root field, enter the host name (for example, **xrv9k-26**) or select a node on the map or an existing device in the list. As you type or select the Root information, a Root label for the selected node appears on the map. Only PCC nodes with PCEP session to PCE can be added as a Root node.

c) In the Leaf field, enter the host name (for example, **xrv9k-24**) or select a node on the map. As you type or select the Leaf information, the Leaf label(s) for the selected nodes appear on the map.

   Click **+ Add another** to add additional constraints (for example, **xrv9k-27**).

d) For the Optimization Objective, select one of the following constraints: Interior Gateway Protocol (IGP) Metric, Traffic Engineering (TE) Metric, or Latency (for example, **IGP**).

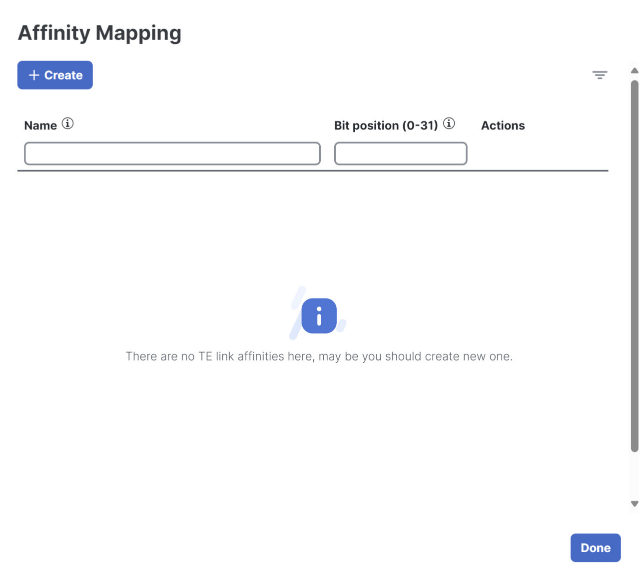
e) For LFA FRR, select **Enable** or **Disable** (for example, **Enable**).

By selecting Enable, the Loop Free Alternate Fast Reroute (LFA FRR) is enabled on all of the nodes in the distribution tree.

f) For additional Constraints, select one of the following Affinity options: **Exclude-Any**, **Include-Any**, **Include-All**.

In addition, from the Select or Create Mapping drop-down list, click **Manage Mapping**. The Affinity Mapping dialog box opens. For more information on Affinities, see the Configure Link Affinities section in the Crosswork Optimization Engine guide.

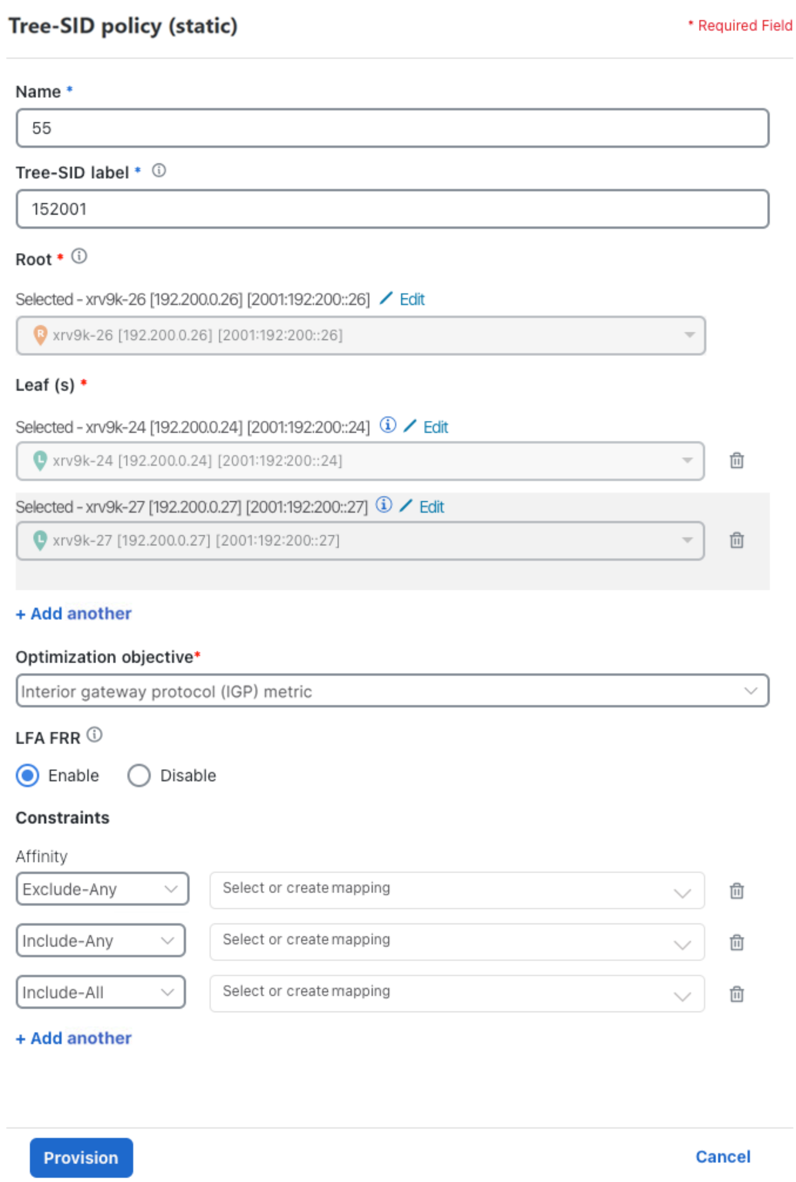
*Figure 109: Affinity mapping*




g) For Affinity Mapping, type the mapping's Name (color) and enter the Bit Position (**0 – 31**). Enter the same bit position as used on the device interface. Click **Done**.

To create additional constraints, click **+ Create**.

*Figure 110: New Tree-SID policy (static) details*




h) To commit the policy, click **Provision** to activate the policy on the network.

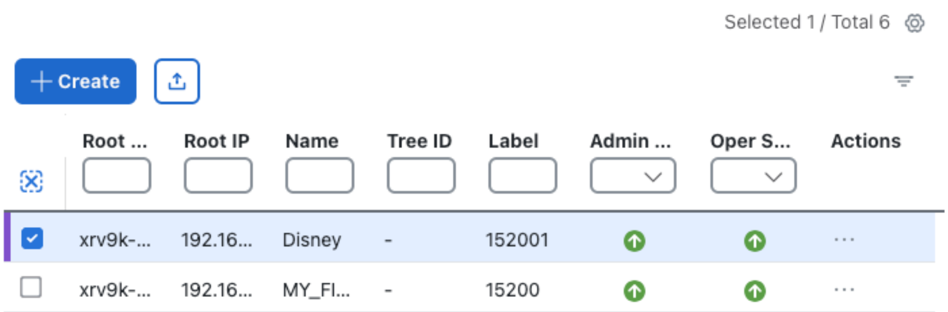
---

The newly provisioned Tree-SID policy may take some time to appear in the Tree-SID table, depending on the network size and performance. The Tree-SID table is auto-refreshed every 30 seconds. Once the request is successful, select **View Tree-SID Policy List** or **Create New** to add additional policies. If you select **View Tree-SID Policy List**, the Tree-SID Policy screen shows the newly created policy in the table.

# Step 2 Visualize and validate the new static Tree-SID policy

**Step 1** Select the root Tree-SID policy check box from the list. In this example, select **xrv9k-26**.

*Figure 111: Tree-SID policy*




If the table contains a large number of policies, filter by Root IP, Name, Label, or other parameter, to help locate the policy you want to visualize.

The map will show the selected Tree-SID policy as an overlay on the topology. It shows a representation of the Tree-SID policy routes, with icon flags indicating the root node (**xrv9k-26**, also known as the ingress device) and the two leaf nodes (**xrv9k-24** and **xrv9k-27**, also known as egress devices), with intermediary transit nodes between them. The administrative and operational status for each node is shown in the table.

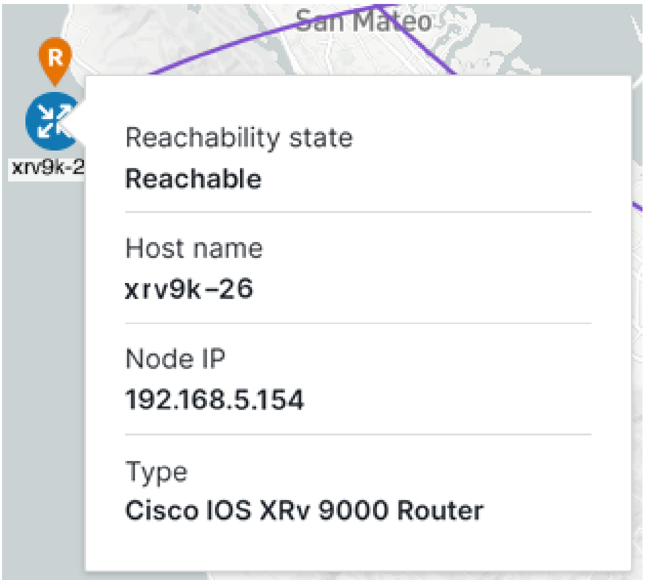
**Note** Use the buttons at the top right of the logical map to toggle between the Logical Map and the Geo Map views.

**Step 2** Select the **Geo Map** button to view the selected Tree-SID service topology overlaid on a world map.

**Step 3** In the map, select the **Show: Participating only** check box to hide underlay devices that are not participating in the selected Tree-SID policy. Then, use your mouse to hover over the **xrv9k-26** root device to view its corresponding Reachability State, Host Name, Node IP, and device Type.

Check any participating Tree-SID device in the same fashion to view their corresponding details.

*Figure 112: Tree-SID device details*




**Step 4** In the map, click **xrv9k-24**.

The Device Details screen opens, showing **xrv9k-24** information organized by Summary and Routing in the Details tab and PCEP Sessions in the Traffic Engineering tab.

*Figure 113: Device details*

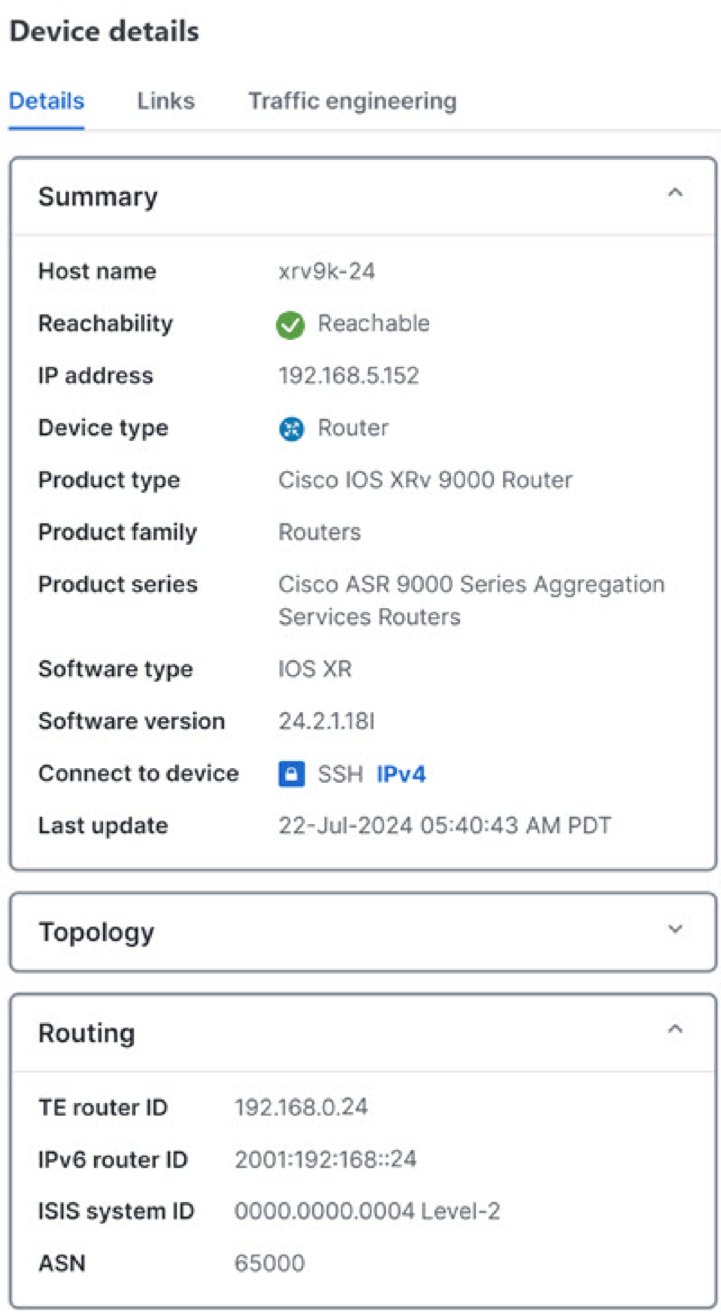
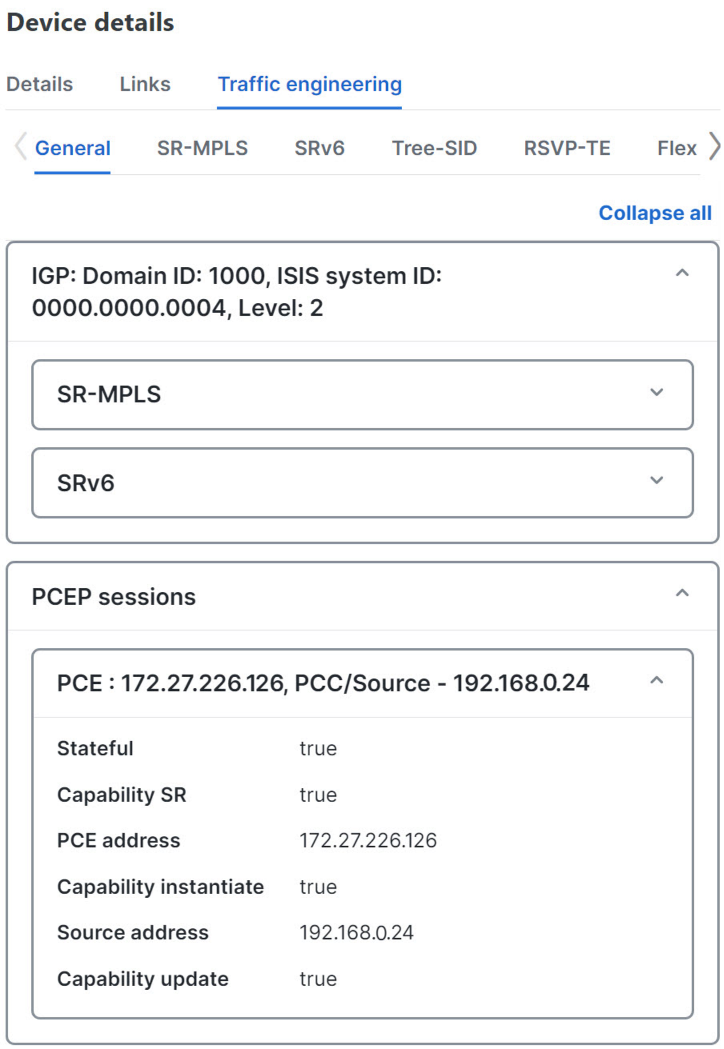


*Figure 114: Device details traffic engineering*




**Step 5** Click **X** in the top-right corner to return to the Tree-SID Policy table to close the Device Details screen and select the **Tree-SID** tab again.

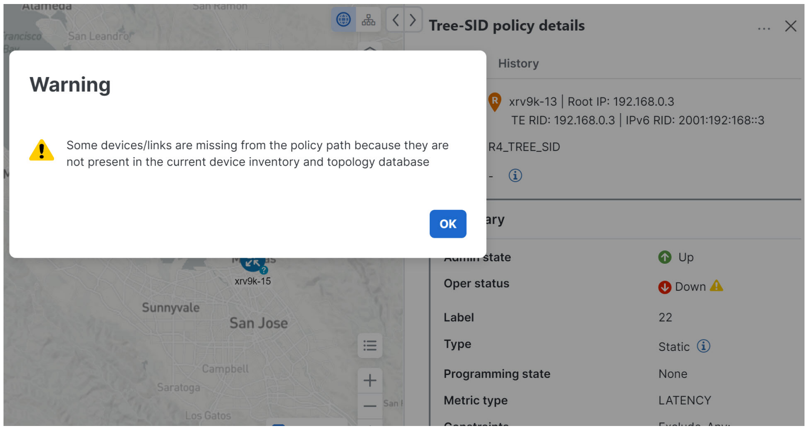
**Step 6** In the Tree-SID Policy list for the selected **xrv9k-26** device, click ⋯ in the **Actions** column and select **View Details** to drill down to a current and detailed view of the Tree-SID policy.

The Tree-SID Policy Details screen appears.

**Note** To view all of the Tree-SID Policy Details, click **See more**.

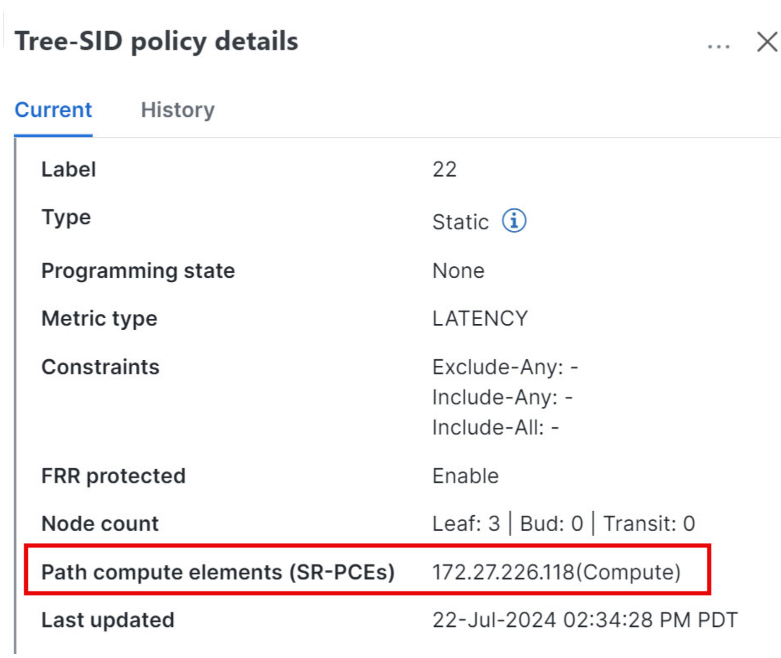
**Note** When viewing Tree-SID Policy Details, if a Source Node is unavailable, a warning message and a warning icon appears next to the Oper Status field, detailing where the connection issue resides. For example:

*Figure 115: Missing source node warning icon/message*




**Note** A (Compute) label, next to the SR-PCE field, details the SR-PCE used to create the policies. For example:

*Figure 116: SR-PCE path compute details*




**Step 7** In the Tree-SID path section, click **Expand All** to view Tree-SID path names and IPs for the **xrv9k-24** and **xrv9k-27** leaf nodes. The list also shows details for the corresponding Root node, all Transit nodes, the two Leaf nodes, and their Egress Link's Local IP and Remote IP information.

**Step 8** Deselect the **xrv9k-22** check box to see Tree-SID path details for **xrv9k-24** and **xrv9k-27** devices only.

The topology updates to show only the selected **xrv9k-24** and **xrv9k-27** Tree-SID routes.

**Step 9** Click **X** in the top-right corner to return to the Tree-SID Policy table.

**Step 10** Select the Root IP Tree-SID policy **xrv9k-26** check box from the list. Make sure the geographical map option is selected. The geographical map updates to show the policy and its disjunct routes. You can click on individual links and get details on the Tree-SID policies in which each link participates.

# Step 3 Associate the static Tree-SID policy with the newly created L3VPN service model

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)**.

The Provisioning screen appears, showing available Services/Policies.

**Step 2** Select **L3VPN > L3vpn-Service**.

The L3VPN > L3vpn-Service table appears.

**Step 3** To create a new L3vpn-Service, click the + symbol.

The Create L3VPN > L3vpn-Service screen appears.

**Note** You may also click the symbol to import an existing L3vpn-Service.

**Step 4** In the vpn-id field, type the unique ID for the service (for example, **MVPN-TREE-SID-119**) and click **Continue**.

**Note** This identifier has a local meaning (such as within a service provider network).

**Step 5** In the vpn-service-topology drop-down list, select **custom** to define the service topology.

**Note** Point-to-point VPN service topology is not supported.

**Step 6** Expand the vpn-instance-profiles section and click the + symbol to add the profile ID.

The vpn-instance-profiles panel appears.

**Step 7** In the profile-id field, type the VPN instance profile identifier (for example, **MVPN-TREE-SID-119**) and click **Continue**.

The vpn-instance-profiles panel refreshes with additional fields to fill.

**Step 8** In the Rd-choice section, enter the directly-assigned rd that indicates an RD value that is explicitly assigned (for example, **0:70:70**).

**Step 9** For address-family, click the + symbol. The address-family panel appears. Select **ipv4** from the address-family list and click **Continue**.

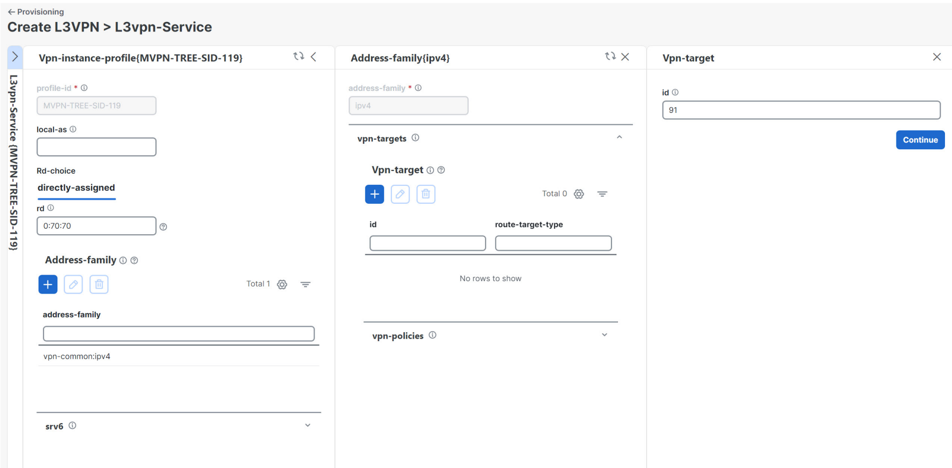
The address-family{ipv4} panel updates with vpn-targets section included.

**Step 10** For vpn-target, click the + symbol so to signify the VPN target id and route-target-type.

The vpn-target panel appears.

**Step 11** In the id field, enter the id (for example, **91**) and click **Continue**.

*Figure 117: L3VPN-service*




**Step 12** In the vpn-target{91} panel, select the route-target-type drop-down list and select **both**.

The address-family{ipv4} panel updates showing the vpn-target id (as **91**) and route-target-type (as **both**).

**Step 13** In the vpn-target{91} panel for route-targets, click the + symbol, type the route-target (for example, **0:70:70**), and click **Continue**. Click **X** to close the panel.

The route-target table updates with the new information. Click **X** in the top right to close all of the remaining panels.

Adding the vpn-instance-profiles is now complete.

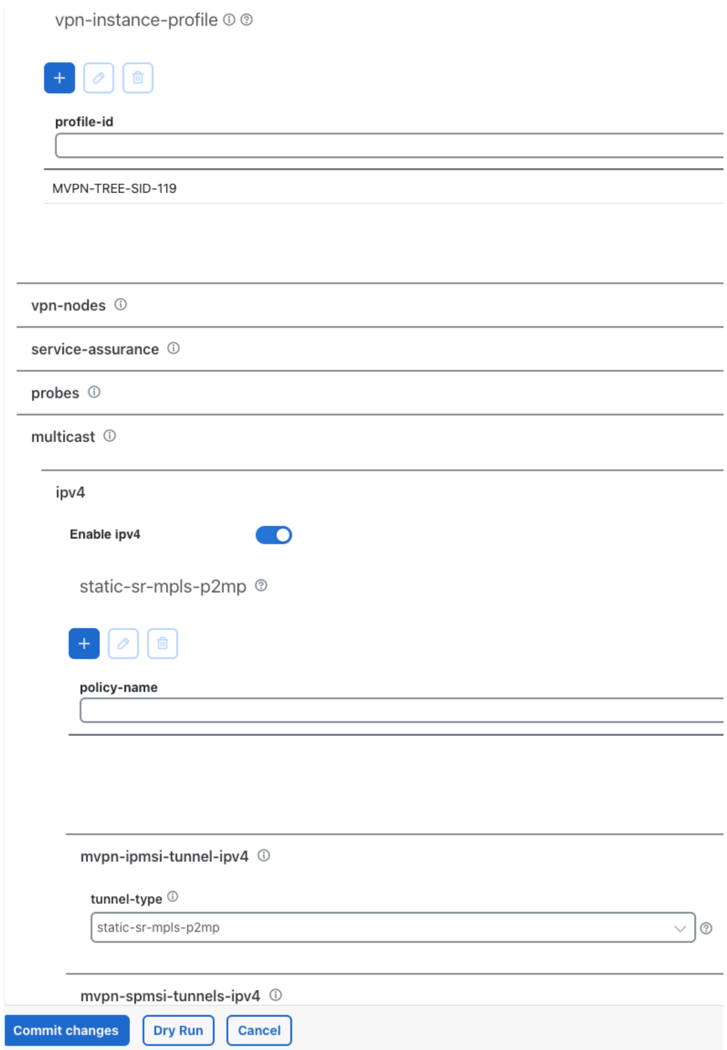
**Step 14** Select **multicast** and then ipv4 to expand both sections.

**Step 15** Expand the mvpn-ipmsi-tunnel-ipv4 section and select **static-sr-mpls-p2mp** from the tunnel-type list.

The Enable ipv4 toggle is now switched on.

**Note** The sr-mpls-p2mp selection in the list is for a Dynamic Tree-SID policy.

*Figure 118: VPN instance profile*




**Step 16** For static-sr-mpls-p2mp, click the symbol.

The static-sr-mpls-p2mp panel appears.

**Step 17** In the policy-name field, type the previously created Static Tree-SID policy name (for example, **xrv9k-26**) and click **Continue**.

The static-sr-mpls-p2mp{Static-xrv9k-26} panel updates.

**Step 18** In the sr-p2mp-policy area for the group-address, click the symbol to add the address.

The group-address panel appears.

**Step 19** In the Address field, type the IPv4 static multicast group address (for example, **1.1.1.1**) and click **Continue**.

The group-address{1.1.1.1} panel refreshes. Click **X** at the top right to close any remaining panels.

**Step 20** Click the symbol in the multicast > ipv4 subsection to add the other policy name.

The static-sr-mpls-p2mp panel appears.

**Step 21** In the policy-name field, type the other previously created Static Tree-SID policy name (for example, **xrv9k-24**) and click **Continue**.

The static-sr-mpls-p2mp{xrv9k-24} panel updates.

**Step 22** In the sr-p2mp-policy area for the group-address, click the + symbol to add the address.

The group-address panel appears.

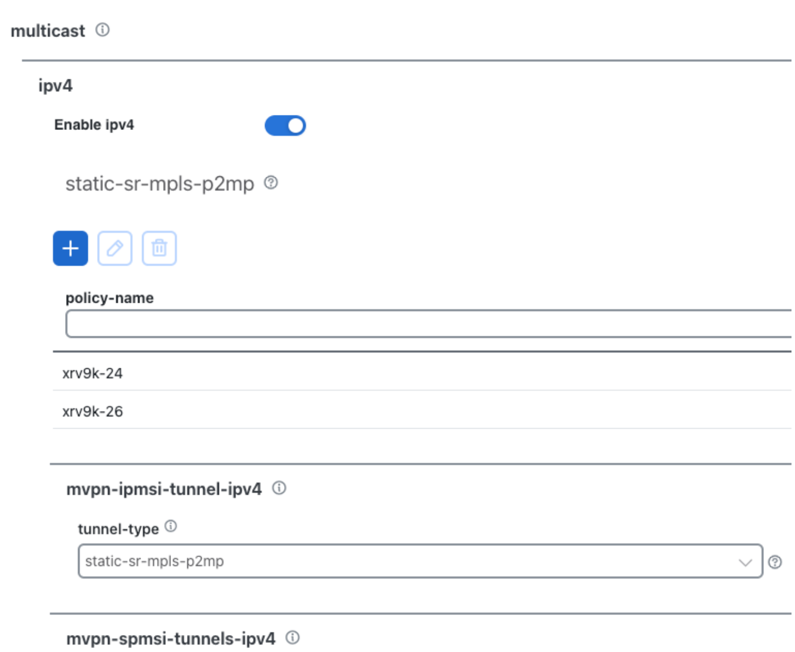
**Step 23** In the address field, type the IPv4 static multicast group address (for example, **2.2.2.2**) and click **Continue**.

The group-address{2.2.2.2} panel refreshes. Click **X** at the top right to close any remaining panels.

You have successfully mapped the static Tree-SID policy to the L3VPN multicast service model. Next, you must add the VPN node details.

**Note** For advanced configurations, you may select mvpn-spmsi-tunnels-ipv4 subsection under the multicast section to define the tunnel-type, switch-wildcard-mode, switch-threshold, per-item-tunnel-limit, group-acl-ipv4 details.

***Figure 119: Multicast section***




# Step 4 Add the VPN nodes

**Step 1** In the vpn-nodes section, click the + symbol to add your VPN nodes set up in the Static Tree-SID policy (**xr9k-26**, **xr9k-24**, and **xr9k-27**).

The vpn-node panel appears so to add the vpn-node-id.

**Step 2** From the vpn-node-id drop down, select the first of the VPN node (for example, **xr9k-26**) and click **Continue**.

The vpn-node{xr9k-26} panel updates with additional fields.

**Step 3** In the Local-as field, type **65000**.

**Step 4** In the active-vpn-instance-profiles section, click the + symbol to add the VPN instance profile ID.

**Step 5** In the profile-id drop down list, the previously added profile ID appears. Select it (for example, **MVPN-TREE-SID-119**), click **Continue** and click **X** to close the panel.

**Step 6** In the vpn-node{xr9k-26} panel, select the vpn-network-accesses section and click the + symbol to add the vpn-network-access ID. In the Id field, add a number (for example, **1**) and click **Continue**.

The vpn-network-access{1} panel updates with additional fields.

**Step 7** In the interface-id field, type the identifier for the physical or logical interface (for example, **Loopback70**).

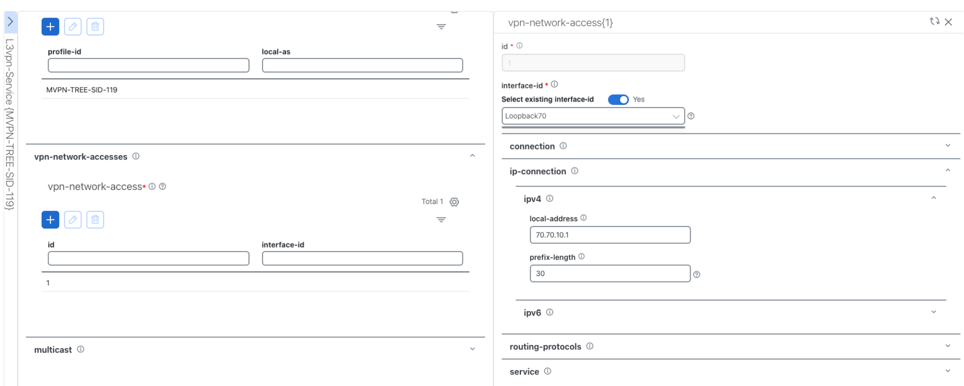
The identification of the sub-interface is provided at the connection level and/or the IP connection level.

**Step 8** In the ip-connection section, select the ipv4 subsection, and in the local-address field, type the IP address used at the provider's interface (for example, **70.70.10.1**).

**Step 9** In the prefix-length field, type **30**.

The subnet prefix length is expressed in bits. It is applied to both local and customer addresses.

***Figure 120: VPN network access***




**Step 10** In the routing-protocols section, click the + symbol to add the unique identifier for the routing protocol. In the Id field, type **bgp** and click **Continue**.

The routing-protocol{bgp} panel appears.

**Step 11** In the type drop-down list, select **bgp-routing**.

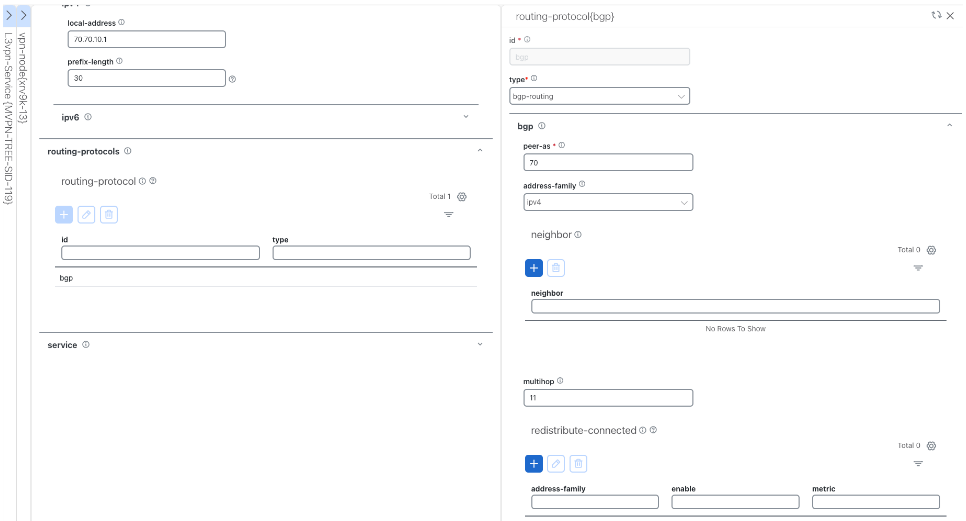
The routing-protocol{bgp} panel refreshes with additional sections.

**Step 12** In the bgp section, for the peer-as field, type **70** to indicate the customer's ASN when the customer requests BGP routing, and in the address-family drop-down list, select **ipv4**. This node contains the address families to be activated.

**Note** If you select dual-stack, it means that both ipv4 and ipv6 will be activated.

**Step 13** In the multihop field, type **11** to describe the number of IP hops allowed between a given BGP neighbor and the PE.

*Figure 121: Routing protocol*




**Step 14** For the neighbor section, click the symbol. In the Neighbor field, type the device address (for example, **70.70.10.2**) and click **Continue**.

**Step 15** For the redistribute-connected section, click the symbol and from the address-family drop-down list, select **ipv4** and click **Continue**.

The redistribute-connected{ipv4} panel appears.

**Step 16** In the enable field, select **true** to enable the redistribution of connected routes.

Close all panels (click X in the top right corner) until the Create L3VPN > L3vpn-Service screen appears.

**Step 17** In the vpn-nodes section, you will see xrv9k-26 listed in the vpn-node table. Select **xrv9k-26** and select the symbol.

The vpn-node{xrv9k-26} panel appears.

**Step 18** Select the multicast section and click the symbol to add the mapping of the policy for each node.

The static-sr-mpls-p2mp panel appears.

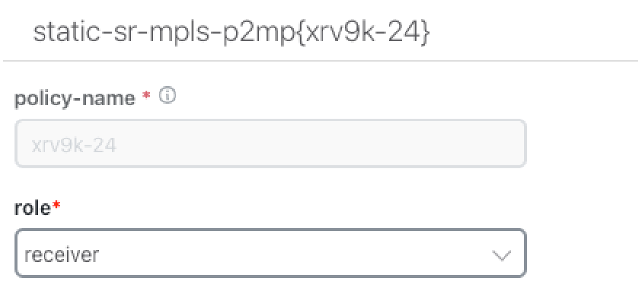
**Step 19** For the policy-name drop-down list, select the policy you want to add to this node (either the source or the receiver). Select **xrv9k-24** as a receiver and click **Continue**.

The static-sr-mpls-p2mp{xrv9k-24} panel updates with additional fields.

**Step 20** For the role drop-down list, select **receiver**.

Close all additional panels (click X in the top right corner) until the Create L3VPN > L3vpn-Service screen appears.

*Figure 122: Policy mapping to the node*




**Step 21** Repeat steps 1 – 20 to add the other two VPN nodes set up in the Static Tree-SID policy: **xr9k-24** and **xr9k-26**.

**Step 22** After all the VPN nodes have been added, click **Commit changes**.

# Step 5 Visualize and edit the static mVPN Tree-SID policy's L3VPN service model

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Provisioning (NSO)**.

The Provisioning screen appears, showing available Services/Policies.

**Step 2** Select **L3VPN > L3vpn-Service**.

The L3VPN > L3vpn-Service table appears.

**Step 3** Locate the newly created L3VPN service ID in the table (**MVPN-TREE-SID-119**), and in the Actions column, click ··· and select **Config View**.

The Configured Data pop-up screen appears.

*Figure 123: Configured data pop-up*


**Step 4** In the Configured Data pop-up screen, review the data configuration and click **Copy To Clipboard** if you want to save a copy, or click **Cancel** to exit.

**Step 5** To view the new Static mVPN Tree-SID policy associated with the L3VPN service model, click the name of the VPN Id in the table or the Actions column, click ⋯ and select **View**.

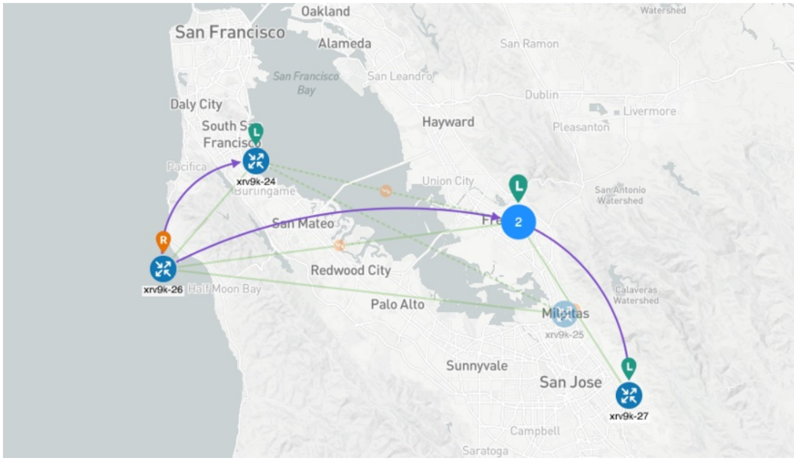
The Service Details screen appears, with a geographical map showing the newly created L3VPN service and its associated nodes: xrv9k-26, xrv9k-24, and xrv9k-27. On the right, the Service Details panel shows the details of the MVPN-TREE-SID-119 service model.

**Step 6** In the Service Details panel, select the **Transport** tab to view the Tree-SID Policy information.

**Step 7** In the table, select the check box next to xrv9k-26.

In the geographical map, the policy will appear showing the one Root, or source, node (xrv9k-26) and the two Leaf, or receiver, nodes (xrv9k-24 and xrv9k-27).

*Figure 124: Policy in geographical map*




**Step 8** Select the second check box next to xrv9k-24.

The geographical map updates.

**Step 9** Use your mouse to hover over the Tree-SID policy names in the table. Depending which policy your mouse hovers over, the geographical map will show the designated path(s) between the nodes so to differentiate them from each other.

**Step 10** For the first policy in the table, in the Actions column, click and select **View details**.

The Tree-SID Policy Details panel appears, showing the policy's details, such as the Name, a Summary section, and the Tree-SID path information, which can be expanded to show additional detail. You may also select the History tab to view historical information for the policy.

**Step 11** To edit or add additional policies, from the main menu, choose **Service & Traffic Engineering** > **Provisioning (NSO)**, and select **L3VPN** > **L3vpn-Service**.

**Step 12** For your L3VPN service, in the Actions column, click and select **Edit**.

The Edit L3VPN > L3vpn-Service screen appears, where you can make additional updates (such as adding VPN nodes to replace a degraded path and give a different route) and modifications to existing details that make up the service.

While editing, to show all or hide the multiple fields that make up the service configuration, select the **Show all fields** toggle at the top right. Click on the toggle to turn it on. Click the toggle again to turn it off, showing just a subset of the fields.

**Step 13** In addition, from the **L3VPN** > **L3vpn-Service** screen, click in the Actions column and select **Edit in Json editor** for your L3VPN service.

The json Configuration editor appears. You can highlight different details that make up the service configuration and edit them directly in the json editor.

*Figure 125: Actions menu options*

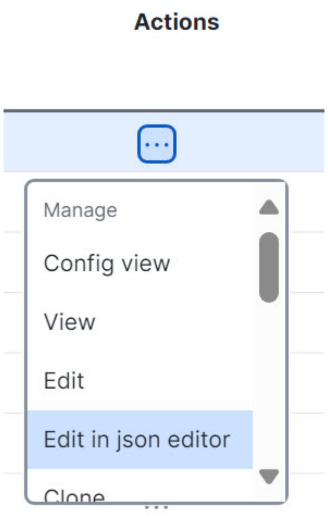



*Figure 126: JSON editor*


**Step 14** Once completed, click **Commit** to initiate the changes and update the service's configuration or click **Cancel**.

# Summary and conclusion

As we observed, you can provision new static Tree-SID policies within the Crosswork Network Controller UI. Once provisioned, you can use the Tree-SID tab and its associated map to visualize Tree-SID defined routes, identify disjunct policy routes, and identify problems with transit nodes, interfaces, and links that may affect traffic from the Root to the Leaf nodes. In addition, once the Tree-SID policies are associated with an L3VPN service model, similar capabilities are at hand to visualize and analyze static Tree-SID policies associated with an L3VPN service model and edit in dynamic ways that improve efficiency, accuracy, and ease of use.

CHAPTER 10

# Transport Slice Provisioning

This section explains the following topics:



## Overview

**Objective**

Simplify transport service provisioning by focusing on the service's SLA intents (the "what" instead of the "how"). This implies a service-oriented view, leveraging the concepts of software-defined networking (SDN).

**Challenge**

Service providers face ever-growing demands from end users for highly customized, flexible network services with very different, sometimes contradictory, service level requirements: support for highly mobile smart cars, ultralow-latency AR and mobile gaming applications, secure machine-to-machine communication in logistics and manufacturing, and so on. Modern software-defined network (SDN) traffic engineering technologists have responded with a host of innovative protocols and features that offer many ways to engineer network traffic to meet these special needs. Crosswork support for these approaches, such as SR-TE services, Tree-SID, and Local Congestion Mitigation, are featured elsewhere in this Guide.

The advent of 5G mobile networking has accelerated this trend, resulting in a new approach to network architecture: network slicing. This still-emerging standard enables network engineers to slice the 5G network's bandwidth into tranches that prioritize some services over others instead of treating it as a single, monolithic network. The network engineer can design each network slice around the needs and intents of its users, allocating speed, latency, throughput, and other resources to each slice as required. CNC delivers a rich and customizable tool set to make deploying these slices easier. When combined with Service Health, it provides the added ability to easily monitor the health of these services. The provider organization can then offer the slice itself as a service, helping to broaden the range of service offerings.

But how to make these services easy to provision? The design and coding of the sophisticated traffic engineering services that underlie network slicing require the skills of experienced network architects and deep knowledge of each provider's existing network infrastructure. Without automation that allows line operators to instantiate the designed slices quickly and easily, network slices might remain a type of custom configuration, achievable only for a small set of important users, instead of scalable commodity providers that can monetize.

This is an evolving standard. Currently, the Crosswork solution only addresses the Transport-level Network Slice Management Functions (NSMF).

### Solution

Crosswork Network Controller offers direct support for network slicing at the OSI transport layer. Using this solution, network engineering experts can design slices around customer intents and add them to a catalog. Network line operators can then pick the slice intent that best meets the customer's needs, specify the slice endpoints, and (where needed) set any custom constraints or options built into the chosen slice.

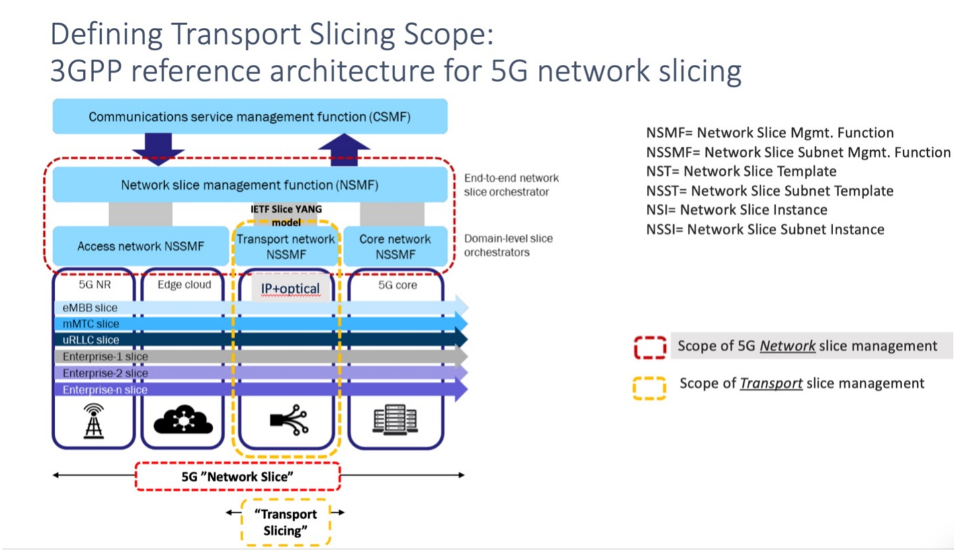
This is Cisco's initial offering in the network slicing arena, chosen because of our company's strengths at the transport layer. Currently, the Crosswork solution provides a large catalog of slice template examples and extensive customization for each template. This document offers a scenario that you can follow to create and (optionally) monitor a network slice.

### How does transport slicing work?

It's important at the outset to understand the difference between 5G network slicing and generalized transport slicing. When operational, a 5G network slice is an end-to-end solution crossing multiple sub-domains. The 5G network authority 3rd Generation Partnership Project (3GPP) refers to each end-to-end network slice operating on the network as the Network Slice Instance (NSI). Each NSI is a unique entity, provisioned in the network with a set of Service Level Requirements chosen from a set of pre-created Network Slice Templates (NST).

All NSIs must be orchestrated by a controller called the Network Slice Management Function (NSFM). The NSMF, in turn, communicates with sub-domain controllers, referred to as Network Slice Subnet Management Functions (NSSMF). Each NSSMF provisions the corresponding domain-specific slice instance across its sub-domain boundaries (called a Network Slice Subnet Instance or NSSI). For the Transport domain, the IETF has defined the NSSI as an “IETF Network Slice” to differentiate slices in the transport domain from slices bridging other domains. The space occupied by transport slicing in this hierarchy is shown in the illustration below, where the CNC solution will provide the NSSMF functionality for the Transport domain. It is important to highlight that Cisco's Transport Slicing solution can be used independently from 5G use cases, as it's a generic solution for implementing any transport service based on intents.

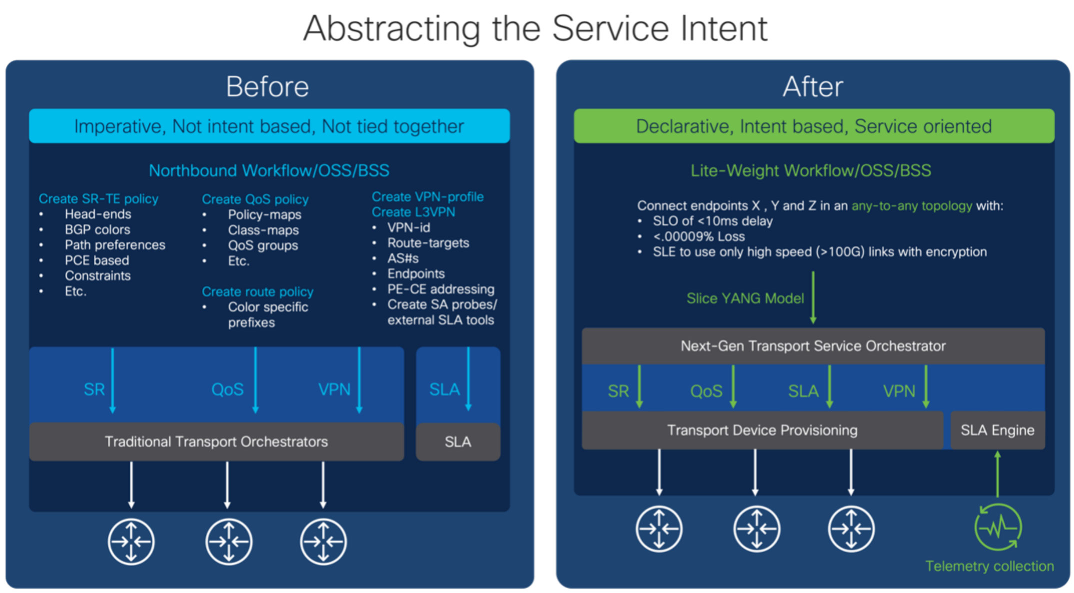
*Figure 127: Defining transport slicing scope*



Simplification and ease of use are key principles in transport slicing. The operator wants to start very simply, by requesting from a controller a service based on a desired service intent or outcome (such as supplying low latency to an AR application). They then want the controller to build the service.

The controller must also monitor the built service to ensure it honors the operator's intent. Above all, the operator wants to avoid exposure to the many configuration parameters required to deploy the service at the device layer. Realizing that vision involves the creation of intent-based modularity for value-added transport services supporting the slice, using well-abstracted and declarative service-layer APIs. These service APIs must be maintained and exposed by a controller that can act in a declarative and outcome-based way, as shown in the following figure.

***Figure 128: Abstracting service intents and outcomes***




Monitoring the slice's fidelity to the intent involves a Service Level Agreement (SLA) between the customer and the slice provider. To ensure that this SLA effectively reflects the slice's intended purpose and includes specific, actionable terms, it can be further categorized as either a Service Level Objective (SLO) or a Service Level Expectation (SLE):

- **Service Level Objective (SLO)**: A desired and achievable target value or range of values for the measurements obtained from observing a Service Level Indicator (SLI). For instance, an SLO may be stated as "SLI <= target" or "lower bound <= SLI <= upper bound".
- **Service Level Expectation (SLE)**: The expression of an unmeasurable service-related request that a customer makes of the provider. An SLE differs from an SLO as the customer may have limited means of ascertaining whether the SLE is being fulfilled but still enters into an agreement with the provider for a service that meets the expectation (refer to the sample SLE table below).

***Table 5: Sample service level expectations***

| SLE | Description |
|---|---|
| Encrypted Link Services | Traffic must transit encrypted links only. |
| Disjoint Path Services | The network has multiple forwarding planes with no common nodes or links. |

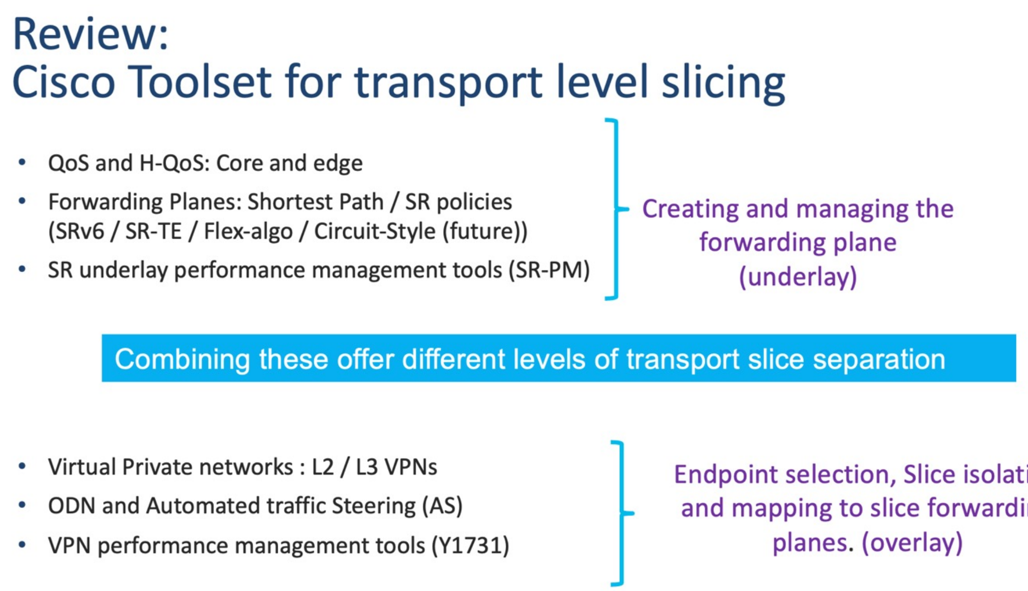
| SLE | Description |
|---|---|
| High speed links only | Traffic must transit high-speed links only. Links offering speeds greater than or equal to 100Gbps are typical for "elephant flows". |
| Lowest Latency | Always take the lowest latency path. No SLO would be specified in this case. |
| Regional Avoidance | Do not use nodes or links in specific regions or countries. |
| Trusted Nodes | Only use trusted nodes ("trusted" meaning verified and not in the common carrier space). |
| L4-L7 Services | Redirect to “in-line” L4-L7 service on traffic (typically used for security services). |
| Reliable Links | Use only transit links that have optical protection and L1 diversity. |
| “Circuit-Style” Services | Provide L1 circuit-like connectivity. |
| Gaming Services | Use network segments optimized for network gamers (low latency, high bandwidth) |
| Connected Car | Use network segments optimized for network-connected cars (low latency, close proximity) |
| Cloud Provider-Specific | Connect me to the secure “walled garden” for a cloud provider (such as AWS or Azure). |

The SLA, therefore, sets key goals and measures to be applied for a given connectivity construct between a sending endpoint and the set of receiving endpoints. It also describes the extent to which divergence from individual SLOs and SLEs can be tolerated and the specific consequences for violating these SLOs and SLEs.

**What makes up a Cisco transport slice?**

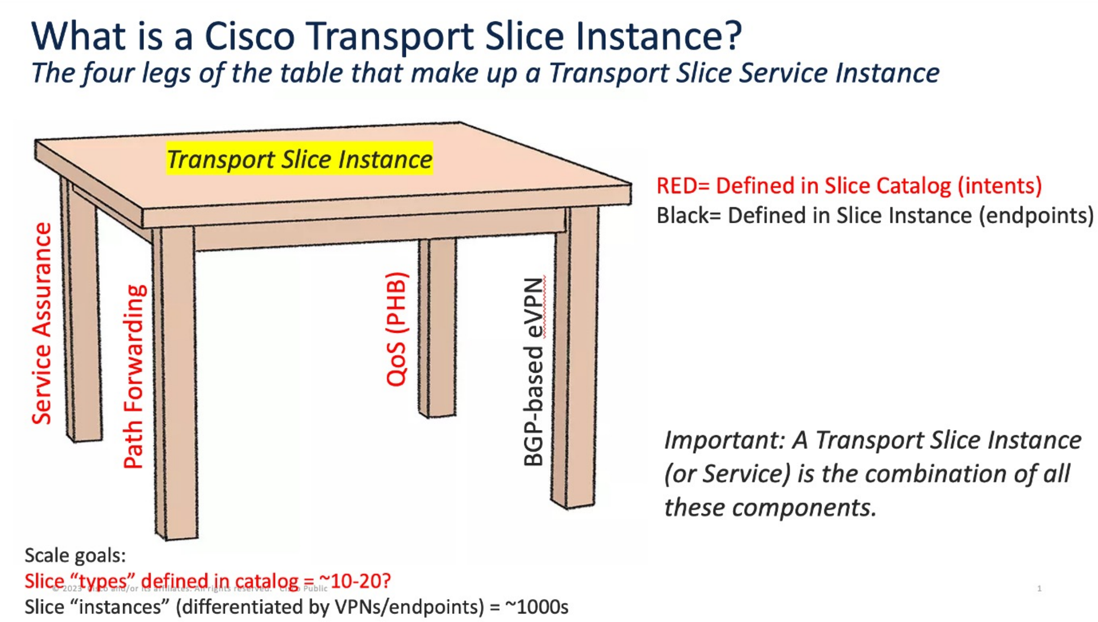
To build and deploy these highly abstract intents, Crosswork Network Controller must translate them into actual device configurations. Governing bodies like the IETF and 3GPP leave these decisions to vendors. Cisco can leverage a complete toolkit built over many years of innovation, as shown in the following figure.

*Figure 129: Cisco's transport slicing toolkit*




For a Cisco Transport Slice Instance, we categorize the features in the preceding illustration as follows: Service Assurance, Path Forwarding, QOS (PHB), and BGP-based EVPN. The configurations in these categories support the slice instance, as shown in the illustration below.

*Figure 130: Cisco's transport slice instance*




The first three of these features (shown in red) are defined in the slice template catalog (this catalog is equivalent to what 3GPP calls the NSST). The slice catalog contains templates, each defined once by a slice designer. Slice *templates* are just blueprints and are not instantiated in the network in any way. Slice *instances* are the instantiated services after they are deployed in the network. The end-user doesn't need to know the details of what is inside the templates, just the overall intent (or SLA) for each slice template. The slice template catalog is thus a catalog of slice intents.

The fourth category – BGP-based VPNs – that make up a Cisco Transport Slice Instance is the selection of endpoints and service types (L2 or L3 forwarding). Operators define these when deploying the Cisco Transport Slice Instance.

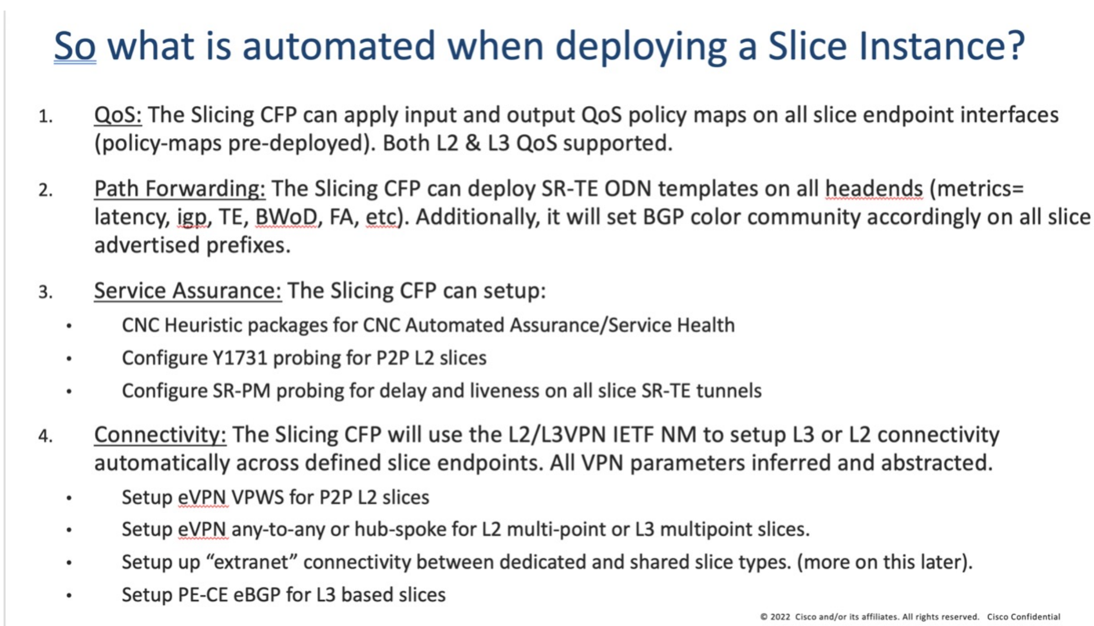
The benefit of this approach is to fully abstract the underlying configuration details of the various machinery components required to realize a Cisco Transport Slice Instance (aka the IETF Network Slice, or, in 3GPP parlance, the Network Subnet Slice Instance or NSSI).

To deploy a new slice instance, the operator executes the following steps:

1. Select a slice intent from the available Templates in the Slice Catalog.
2. Select slice endpoints and connectivity details, which drive the VPN configuration. Once committed, Crosswork Network Controller will then provision:
   - The forwarding plane policy details which drive the segment routing traffic engineering (SR-TE) configurations and BGP prefix coloring for ODN/AS.
   - The QoS profile details, which drive the ingress marking (for PHB treatment) and the egress scheduling.
   - The SLA details, which will drive the needed Service Assurance configurations.
   - The BGP based VPN connectivity requirements to provide endpoint connectivity.

The following illustration provides more detail on the parts of the slice template that automate slice instantiation.

*Figure 131: What is automated when deploying a slice instance*




### Transport slice high level workflow

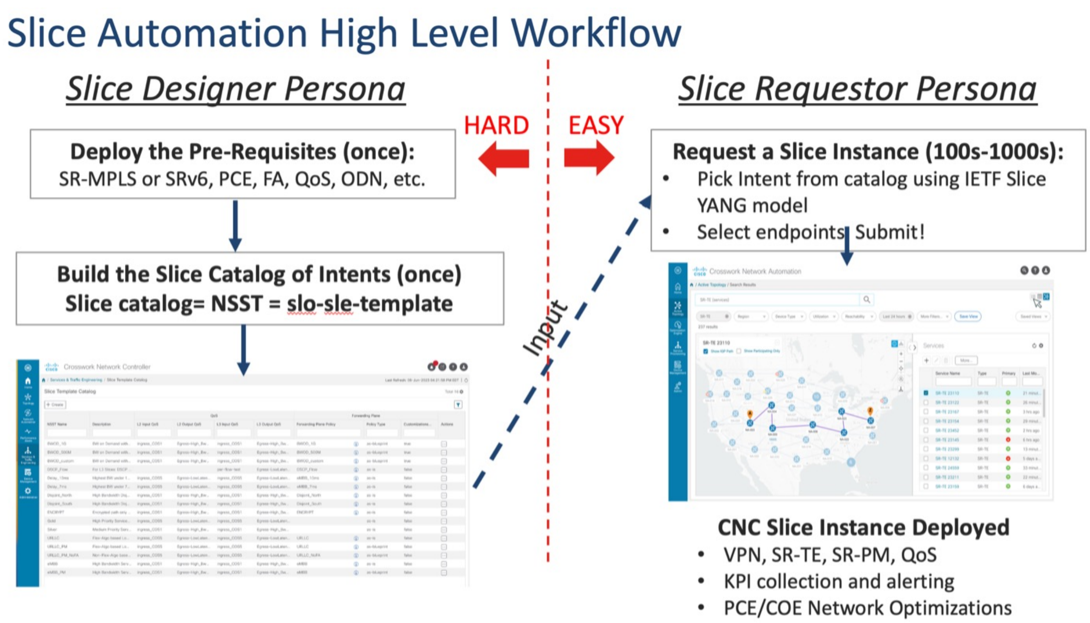
Transport slicing in Crosswork Network Controller is designed around two main user personas:

**Slice Designer**: The designer understands the service requirements the provider organization wants to offer customers and is familiar with the provider network's underlying capabilities. This person has authorization to do one-time setup operations within the network and has a networking engineering background. They will set up the needed network prerequisites and build the slice template catalog, which lists available slice service offerings for network operators.

**Slice Requestor**: The requestor requests new slice instances using the intent-based and simplified CNC user interface. They pick their desired slice type from the pre-built slice catalog, select their endpoints and transport options, and then click submit.

Cisco's objective in Crosswork Network Controller transport slice solution is to make the user experience as simple as possible for the Requestor. This is the only slice deployment operation driving network service provisioning, and as it must be done constantly for a large SP network, it is a major contributor to provider OPEX. The slice template catalog creation is done once by highly skilled designer personnel. While the design step is not automated, this approach leverages those skilled resources in a way that maximizes their value to the provider organization at a scale that cannot be realized if the designer must instantiate every slice by hand. The catalog creation requires a good understanding of the network and its capabilities, and requires pre-requisite configurations as shown in the figure below. Slice designers must be familiar with all the pre-requisite configuration types listed in the illustration for this approach to work.

*Figure 132: Role of the slice designer and requestor*




# Scenario: Implement an any-to-any L3 eMBB slice

In this scenario, you require a transport slice with Layer3 any-to-any connectivity across three endpoints, using the intent defined in the catalog for Mobile Broadband (eMBB). The eMBB intent will provide the highest bandwidth available path (including proper QoS marking/scheduling treatment), and some basic service assurance capabilities such as endpoint interface status and PE-CE route health. The eMBB intent will also enable you to specify:

- The highest bandwidth available path.

- Some basic service assurance capabilities.
- eBGP peer routing connectivity details to CE devices.

### Assumptions and prerequisites

This scenario assumes the network has already built the required network capabilities for this intent. However, they will be briefly reviewed here for this scenario. For more detailed explanations, see the Cisco Transport Slice Automation Design Guide.

#### Slice service package prerequisites

The NSO slice services package uses a few optional prerequisites that need to be bootstrapped into NSO. The prerequisites required depend on the types of slices and intents required.

First, suppose you plan on using any “as-blueprint” forwarding-plane-policy-types in your template catalog. In that case, you must create an NSO sr-color resource pool so that the slicing service can dynamically assign colors to create ODN policies. This pool should then be referenced by the slicing service.

Second, if creating point-to-point L2 slice service types, the route-policy map assigned to the BGP session for the route reflector must be identified to the slicing package. This policy map will be modified by the slicing package with new policies as needed for L2 services, which is a standard approach for VPWS.

In this scenario, these items are not required since you are using neither of these functions:

```
resource-pools id-pool sr-color-pool
range start 1000
range end 2000
!
network-slice-services cfp-configurations color-pool-name sr-color-pool
network-slice-services global-settings parent-rr-route-policy SET_COLOR_EVPN_VPWS
!
```

#### Path forwarding prerequisites

The following settings have been preconfigured with the NSO T-SDN SR-TE CFP for the eMBB ODN path-forwarding intent with these properties:

- Use Color 100 to identify the intent.
- PCE is responsible for dynamic path computation
- The dynamic path computation will be based on the IGP metric.

On NSO, this set of properties will look like the below example. At this stage, the ODN policy has not been pushed to the devices.

```
admin@ncs# show running-config cisco-sr-te-cfp:sr-te odn odn-template eMBB
cisco-sr-te-cfp:sr-te odn odn-templatee eMBB
 color 100
 dynamic pce
 dynamic metric-type igp
!
```

#### QoS prerequisites

As described, you (the Slice Designer) should have a good understanding of the network's settings and device capabilities. You should have a well-designed and implemented QoS design throughout your network. For instance, for QoS treatment for high bandwidth business services, you have chosen to deploy these services with the network's existing "Class of Service 1" traffic policy (called "ingress_COS1").

The details of this policy are again provider-specific, but in this example, the policy will not examine or modify the ingress traffic's IP DSCP setting but mark all the traffic with an MPLS experimental bit (EXP) of 1 so that downstream core scheduling can provide the proper BW treatment. On egress from the provider network, you have chosen a policy called “Egress-High_BW-Apps,” which will assign 50% of the bandwidth to Class of Service 1 (COS1) marked traffic.

It is assumed these QoS policies are already deployed on all edge PE devices (they still need to be added to the customer-facing interfaces but have been built out and ready for use). However, you still need to identify that these QoS policies are available in the Slice Template catalog for Slice services. You will need to provide that mapping and identify which policies are available for either Layer 2 or Layer 3 slice services or both. Since QoS policies can be tailored specifically to Layer 3 or Layer 2 traffic (for example, matching on L3 DSCP vs L2 ToS bits), the system allows you to specify the usage. In the case of the example above, since all traffic is marked with EXP=1 regardless of DSCP or ToS, these policies apply to L2 or L3 services.

***Figure 133: QoS prerequisites***

```
admin@ncs# show running-config network-slice-services slo-sle-templates qos-catalog
network-slice-services slo-sle-templates qos-catalog L2 output-qos-policy Egress-High_Bw_Apps
 description "High BW egress"
!
network-slice-services slo-sle-templates qos-catalog L2 input-qos-policy ingress_COS1
 description "Treat all as Business_Data"
!
network-slice-services slo-sle-templates qos-catalog L3 output-qos-policy Egress-High_Bw_Apps
 description "High BW egress"
!
network-slice-services slo-sle-templates qos-catalog L3 input-qos-policy ingress_COS1
 description "Treat all as Business Data"
!
```

### Slice service assurance settings

In this scenario, you only have basic service assurance requirements, which are based on passive state monitoring (no active probing). You will be using Crosswork Network Controller's Service Health capability and the Crosswork Network Controller system’s pre-built heuristic packages (ConfigProfiles and Rules) which define the objects to be monitored. In the scenario, you want to monitor basic device health and PE-CE route health, which are included in the basic system package. When you build the slice catalog, you can define which packages to use for L2 point-to-point, L2 multipoint, and/or L3 services.

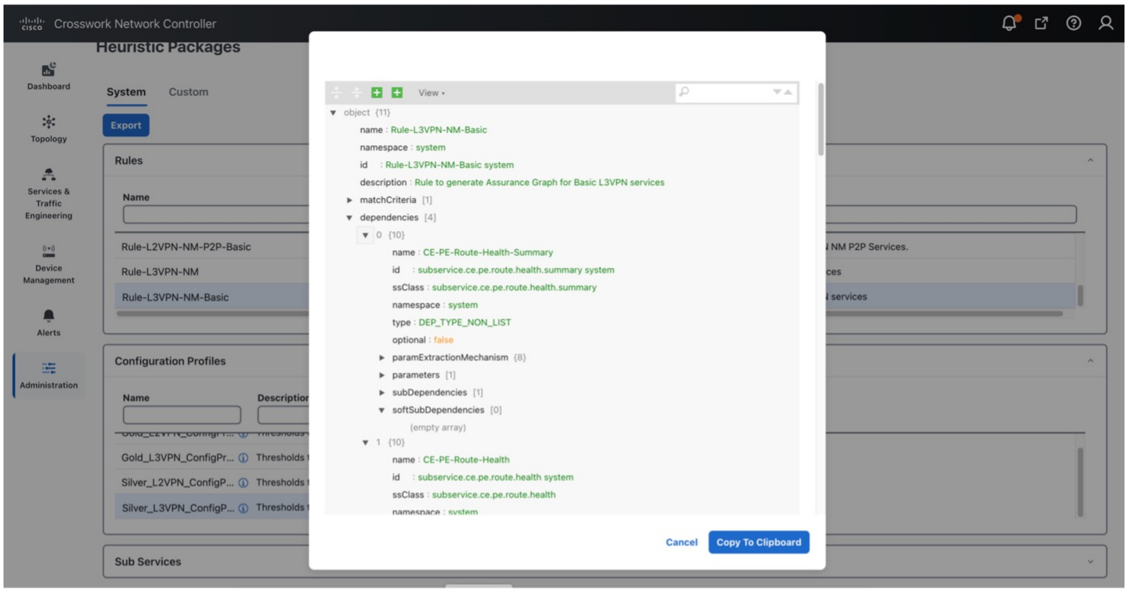
The scenario above requires L3 services, but when you create your catalog for the eMBB intent (next step), you also need to consider future slice instances for eMBB services that are L2 service types. Thus, you can include all these pre-built system heuristic packages in the catalog entry for eMBB. Since these are all pre-built system packages, no prerequisite configurations are required. The system heuristic packages can be viewed via the CNC UI by selecting **Administration> Heuristic Packages** (see figure below).

***Table 6: Heuristic profile and rule names for eMBB***

| System heuristic profile and rule names used for eMBB | Usage |
|---|---|
| Silver_L3PN_ConfigProfile system<br>Rule-L3VPN-NM-Basic system | L3 profile-name<br>L3 rule-name |
| Silver_L2PN_ConfigProfile system<br>Rule-L2VPN-MP-Basic system | L2 multipoint profile-name<br>L2 multipoint rule-name |

| System heuristic profile and rule names used for eMBB | Usage |
|---|---|
| Silver_L2PN_ConfigProfile system<br>Rule-L2VPN-NM-Basic system | L2 point-to-point profile-name<br>L2 point-to-point rule-name |

*Figure 134: Slice service assurance settings*




# Step 1 Create a slice template catalog entry

You will now build out the slice catalog entry for eMBB intent-based slice services. This operation is done once, using the above components as input, and can cover both L2 and L3 slice instance requests. Once complete, you can move to deploy the slice service instance for this scenario, and this entry will now be available for future slice instances requiring eMBB intent services (for example, all of the above prerequisite steps will not be required).

This step, performed by the Slice Designer, builds a slice catalog of intents or slice types that will be referred to when creating the actual slice instance. This catalog (along with the slice instances themselves) can be built in multiple ways:

- Using the Crosswork Network Controller UI
- Using the Crosswork Network Controller or NSO Slicing API
- Using the NSO CLI, including load merge from a text file

This scenario has a Service Assurance option that can only be created using NSO CLI or Crosswork Network Controller API. It will be shown in the Step 2 Add service assurance into the slice template catalog using the NSO CLI (optional)

*Figure 135: Slice template catalog prerequisites*

```
admin@ncs# show running-config network-slice-services slo-sle-templates slo-sle-template eMBB
network-slice-services slo-sle-templates slo-sle-template eMBB
 template-description "High Bandwidth Service with basic SLA monitoring"
 qos-policy L2 input-policy ingress_COS1
 qos-policy L2 output-policy Egress-High_Bw_Apps
 qos-policy L3 input-policy ingress_COS1
 qos-policy L3 output-policy Egress-High_Bw_Apps
 odn forwarding-plane-policy eMBB
 odn forwarding-plane-policy-type as-is
 service-assurance heuristics monitoring-state enable
 service-assurance heuristics L2 point-to-point profile-name "Silver_L2VPN_ConfigProfile system"
 service-assurance heuristics L2 point-to-point rule-name "Rule-L2VPN-NM-Basic system"
 service-assurance heuristics L2 multipoint profile-name "Silver_L2VPN_ConfigProfile system"
 service-assurance heuristics L2 multipoint rule-name "Rule-L2VPN-MP-Basic system"
 service-assurance heuristics L3 profile-name "Silver_L3VPN_ConfigProfile system"
 service-assurance heuristics L3 rule-name "Rule-L3VPN-NM-Basic system"
!
```

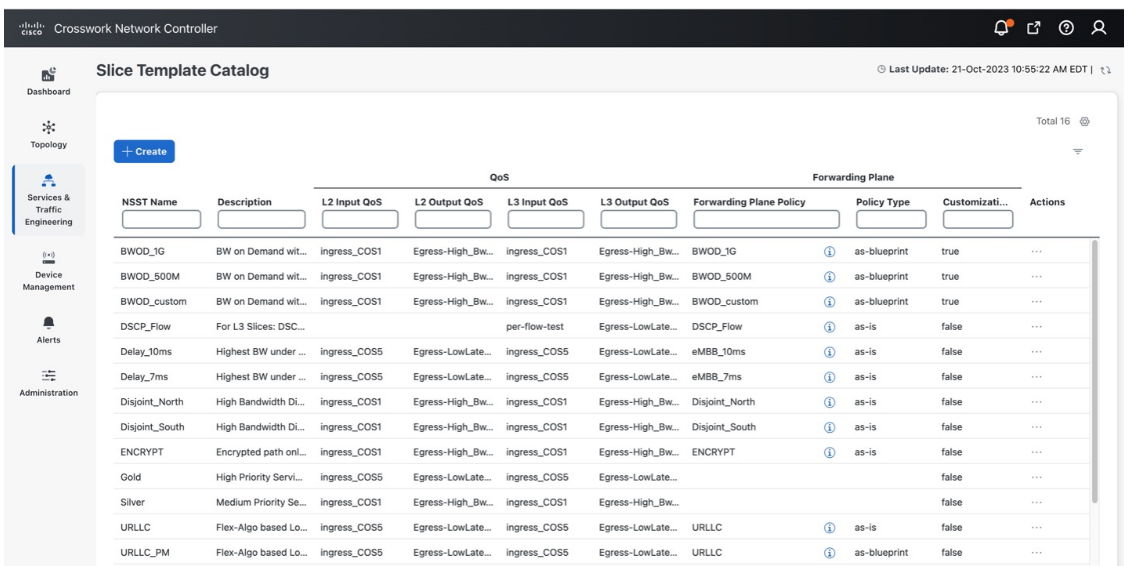
To create a slice template catalog entry using the Crosswork Network Controller UI, do the following:

**Step 1** From the main menu, choose **Services & Traffic Engineering > Slice Template Catalog**.

The Slice Template Catalog screen appears.

**Note** For the purpose of this scenario, the templates that appear in the image below have already been created on NSO.

*Figure 136: Slice template catalog*




**Step 2** Click **+ Create** to create a new slice catalog entry. The New Slice Template screen appears.

**Step 3** For Network Subnet Slice Template (**NSST**), type the new slice template name: **eMBB**. In addition, in the Description field, type a short description of the slice template's intent: **High Bandwidth Service**.

**Step 4** Assign the QoS ingress and egress policies. This depends on the slice instance Service Type (L2 or L3 policy), which you will define later when creating a new slice instance.

**Note** For this scenario, the five fields at the bottom of the screen (L2 Input QoS, L2 Output QoS, L3 Input QoS, L3 Output QoS, Forwarding Plane Policy Template) have already been provisioned in NSO and automatically appear as an option in each list.

**Note** You may also refer to the table built earlier and found under QoS in the prerequisites.

- For L2 Input QoS, select **ingress_COS1**
- For L2 Output QoS, select **Egress-High_Bw_Apps**
- For L3 Input QoS, select **ingress_COS1**
- For L3 Output QoS, select **Egress-High_Bw_Apps**

**Step 5** For the Forwarding Plane Policy Template, select the ODN template policy created earlier (see prerequisites and assumptions section) that complements the forwarding plane intent. For this scenario, select **eMBB**.

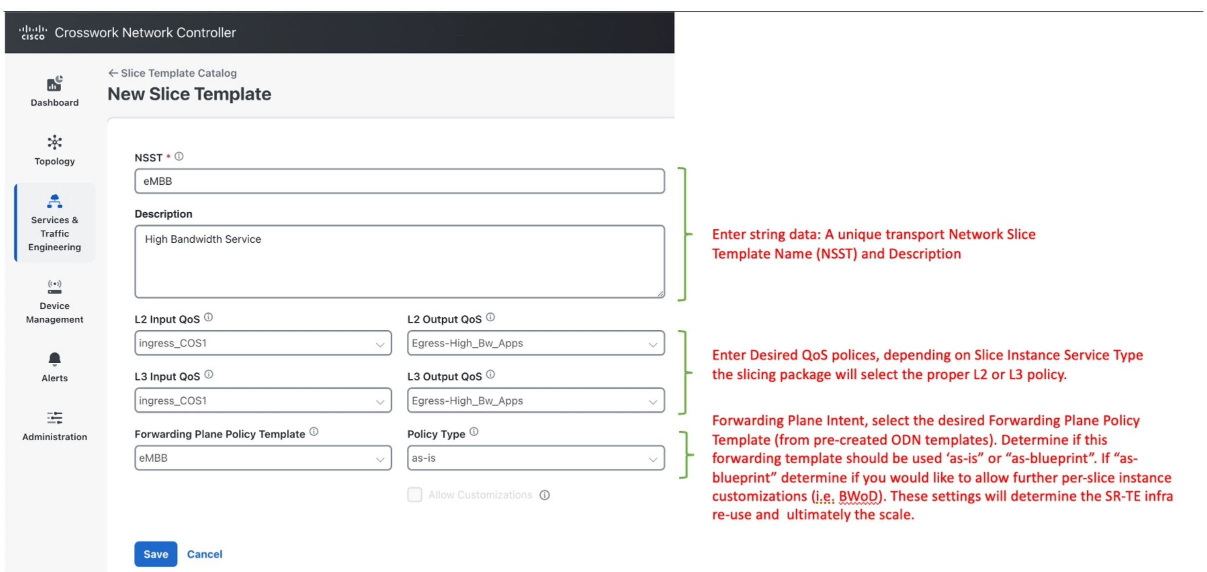
**Step 6** For Policy Type, determine if this template will be used **as-is** or **as-blueprint**. For this scenario, select **as-is**.

The **as-is** forwarding templates increase overall scalability as the SR-TE tunnels can be shared across multiple slice templates and instances. However, these ODN templates will not be dynamically modified by the slice package with additional functionality, including dynamic support for Performance Measurement or BWoD reservations.

**Note** When **as-blueprint** is selected, determine if you also want to allow further per-slice instance customizations. These settings determine the SR-TE infrastructure re-use and the scale. Once **as-blueprint** is selected for Policy Type, the **Allow Customizations** check box will become available.

**Step 7** Click **Save**.

*Figure 137: New slice template*



# Step 2 Add service assurance into the slice template catalog using the NSO CLI (optional)

If Service Assurance is required for the Slice instance, then the Slice Designer can add the necessary Service Assurance functionality into the Slice Template using NSO CLI or API. Below is a template in NSO CLI with Service Assurance parameters to be used for this scenario. It can be added directly into NSO CLI or with the API.

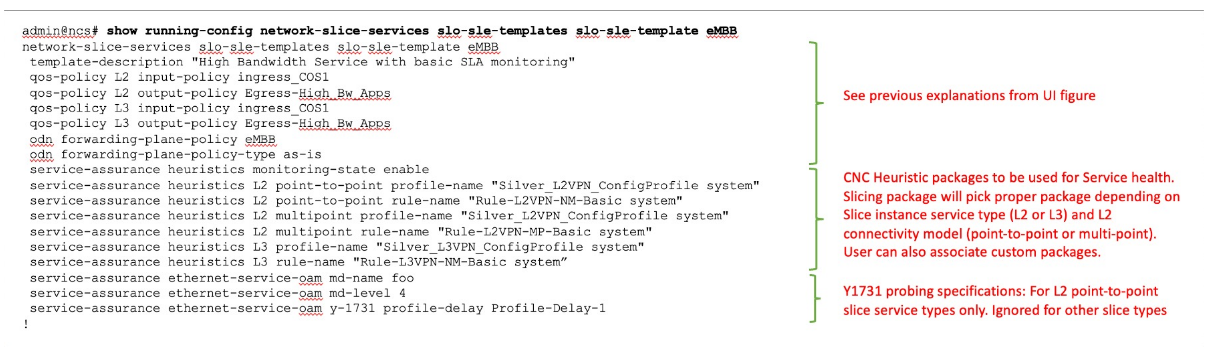
There are three service assurance sections in the template settings:

**Step 1** Reference pointers to Crosswork Network Controller Service Health Heuristic packages to be used and monitoring state. This monitoring state cannot be changed at the Slice instance level at this time; it is set universally for all slice instances referencing this Slice template. Various connectivity types (pt-2-pt or multi-point) can be chosen when setting up a slice instance, as well as different service types (L2 or L3). This means there are multiple heuristic package options available, and the system will choose the appropriate package based on the requirements of the slice instance.

**Step 2** If the Slice Instance is an L2 service type with pt-2-pt connectivity, then Y1731 probe monitoring can be enabled. The required settings are shown in the example below. Slice SLA alarming, and alerting can be configured if the proper settings are selected in the L2 Heuristic package for Service Health.

**Step 3** This scenario does not require Performance Measurement, so it is not included in the template below. However, if desired, it can be enabled in the template, and SR-PM will be dynamically configured on the SR-TE tunnel if the Slice Forwarding policy-type is set for **as-blueprint**. Slice SLA alarming, and alerting can be configured if the proper settings are selected in the Heuristic package for Service Health.

***Figure 138: Slice template catalog***



```
admin@ncs# show running-config network-slice-services slo-sle-templates slo-sle-template eMBB
network-slice-services slo-sle-templates slo-sle-template eMBB
 template-description "High Bandwidth Service with basic SLA monitoring"
 qos-policy L2 input-policy ingress_COS1
 qos-policy L2 output-policy Egress-High_Bw_Apps
 qos-policy L3 input-policy ingress_COS1
 qos-policy L3 output-policy Egress-High_Bw_Apps
 odn forwarding-plane-policy eMBB
 odn forwarding-plane-policy-type as-is
 service-assurance heuristics monitoring-state enable
 service-assurance heuristics L2 point-to-point profile-name "Silver_L2VPN_ConfigProfile system"
 service-assurance heuristics L2 point-to-point rule-name "Rule-L2VPN-NM-Basic system"
 service-assurance heuristics L2 multipoint profile-name "Silver_L2VPN_ConfigProfile system"
 service-assurance heuristics L2 multipoint rule-name "Rule-L2VPN-MP-Basic system"
 service-assurance heuristics L3 profile-name "Silver_L3VPN_ConfigProfile system"
 service-assurance heuristics L3 rule-name "Rule-L3VPN-NM-Basic system"
 service-assurance ethernet-service-oam md-name foo
 service-assurance ethernet-service-oam md-level 4
 service-assurance ethernet-service-oam y-1731 profile-delay Profile-Delay-1
!
```



As previously highlighted, slice templates with service assurance parameters can only be created using the NSO CLI or Crosswork Network Controller/NSO API at this time. This also means these additional parameters will not be visible when viewing the slice template in Crosswork Network Controller UI.

# Step 3 Create the transport slice instance

Once the slice type catalog has been created, we can now deploy the transport slice instance. The table below outlines the user data required to deploy this slice. The mandatory data consists of a series of string-data names (user-defined), selection of the service type (L2 or L3), catalog intent selection, and then defining the Service Demarcation Points (SDPs), which are the PE endpoints facing the customer. These PE endpoints will require

IP information since this is an L3 slice, and optionally, since eBGP was desired for the PE-CE peering protocol, the CE eBGP information is required.

For this scenario, use the sample data below:

***Table 7: Required parameter values***

| Parameter | User value | Mandatory | Notes |
|---|---|---|---|
| slice-service-name | a_L3_A2A_ded | Y | String. Maximum 17 characters. Must be unique. |
| description | “any string data” | N | Any string |
| customer | ACME | N | String meta data- user defined |
| service-tag | L3 | Y | L2 or L3 forwarding |
| nssai | 123459876 | N | String meta data- could match 5G nssai assignment if provider desires |
| slo-sle-template | eMBB | Y | Selection from pre-built slice catalog |
| isolation | dedicated | N | The default is dedicated- the other option is shared |
| **First SDP endpoint name** | 1 | Y | String- unique within slice instance- At least one SDP must be created, the rest optional |
| Node-Name | Node-4 | Y | PE Node-Name as defined in CNC topology |
| Attachment-circuit name | ac1 | Y | String- unique within slice instance |
| Interface-ID | TenGigE0/0/0/10 | Y | Customer facing PE Interface |
| VLAN ID | 401 | N | VLAN ID if using vlan sub-interfaces |
| Interface IP | 172.16.2.1 | Y | PE Interface IP address (since L3 service) |
| Interface IP Mask | 29 | Y | Interface prefix length (i.e. /29) |

| Parameter | User value | Mandatory | Notes |
|---|---|---|---|
| Peering protocol | BGP | N | PE-CE peering protocol (BGP or none) |
| BGP Neighbor ASN | 65102 | Y | Since bgp selected, peer ASN |
| BGP Neighbor Address | 172.16.2.2 | Y | Since bgp was selected, peer IP address |
| **Second SDP endpoint name** | 2 | N | Additional SDPs are optional, but in this scenario we have three endpoints |
| Node-Name | Node-5 | | PE Node-Name as defined in CNC topology |
| Attachment-circuit name | Ac2 | | String- unique within slice instance |
| Interface-ID | TenGigE0/0/0/2 | | Customer facing PE Interface |
| VLAN ID | 301 | | VLAN ID if using vlan sub-interfaces |
| Interface IP | 172.16.1.1 | | PE Interface IP address (since L3 service) |
| Interface prefix length | 29 | | Interface prefix length (i.e. /29) |
| Peering protocol | bgp | | PE-CE peering protocol (bgp or none) |
| BGP Neighbor Address | 172.16.1.2 | | Since bgp was selected, peer IP address |
| BGP Neighbor ASN | 65101 | | Since bgp selected, peer ASN |
| **Third SDP endpoint name** | 3 | N | Additional SDPs are optional, but in this scenario, we have three endpoints |
| Node-Name | Node-2 | | PE Node-Name as defined in CNC topology |
| Attachment-circuit name | Ac3 | | String- unique within slice instance |

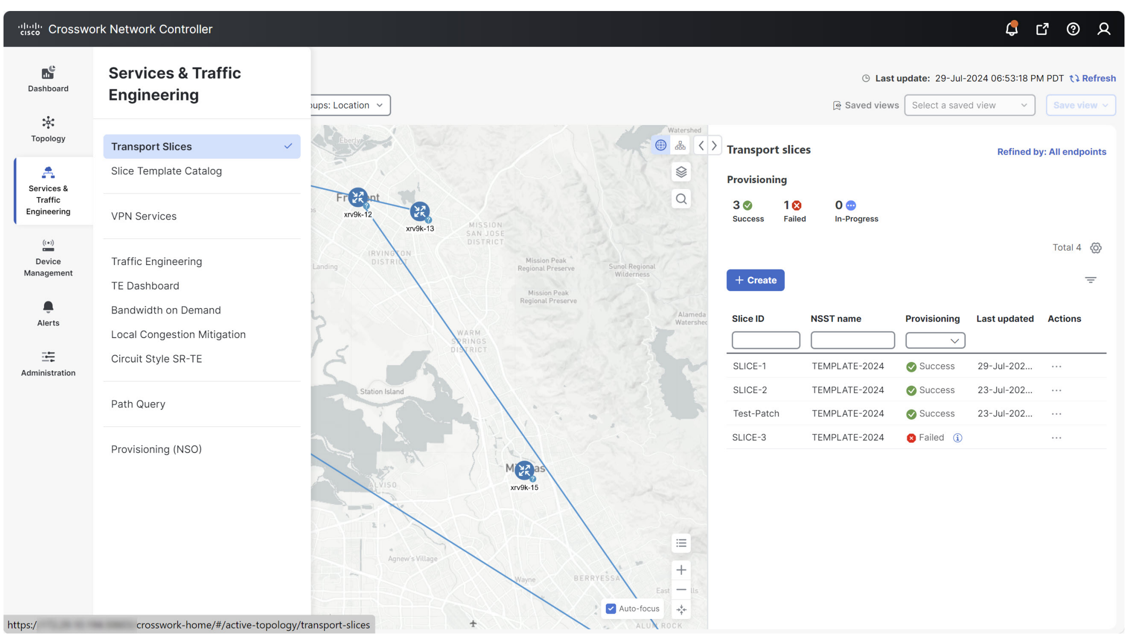
| Parameter | User value | Mandatory | Notes |
|---|---|---|---|
| Interface-ID | TenGigE0/0/0/2 | | Customer facing PE Interface |
| VLAN ID | 601 | | VLAN ID if using vlan sub-interfaces |
| Interface IP | 172.16.3.1 | | PE Interface IP address (since L3 service) |
| Interface prefix length | 29 | | Interface prefix length (i.e. /29) |
| Peering protocol | bgp | | PE-CE peering protocol (bgp or none) |
| BGP Neighbor Address | 172.16.3.2 | | Since bgp was selected, peer IP address |
| BGP Neighbor ASN | 65103 | | Since bgp selected, peer ASN |

The Slice Instance may be created using the Crosswork Network Controller UI, Crosswork Network Controller /NSO API, or NSO CLI. The example below demonstrates the UI steps and explains various fields (both required and optional).

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Transport Slices**.

The Transport Slices panel appears.

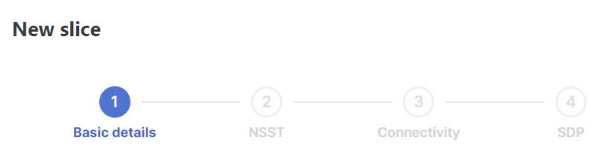
***Figure 139: Transport slicing screen***



**Step 2** Click **+ Create** to create a new slice.

The New Slice panel appears. At the top, four steps are displayed that track the creation of a new slice. The first step requires Basic Details of the new slice.

*Figure 140: New slice steps*




**Step 3** Type the string data into the Slice ID, Customer, and Description fields. For example:

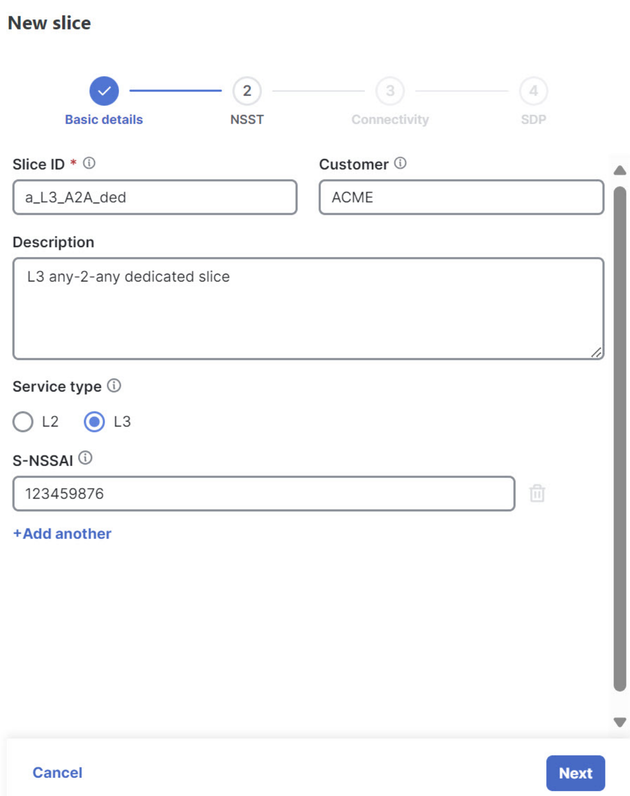
- Slice ID: **a_L3_A2A_ded**
- Customer: **ACME**
- Description: **L3 any-2-any dedicated slice**

**Step 4** Select the Service Type: either Layer 2 (**L2**) or Layer 3 (**L3**) connectivity services. In this instance, select **L3**.

**Step 5** Optionally, add a string-based Single-Network Slice Selection Assistance Information (S-NSSAI) for 5G mobility customers. This mobility slice-ID information is only used as meta-data by the orchestration system. For example, type **123459876**.

**Step 6** Click **Next**.

*Figure 141: New slice - basic details*

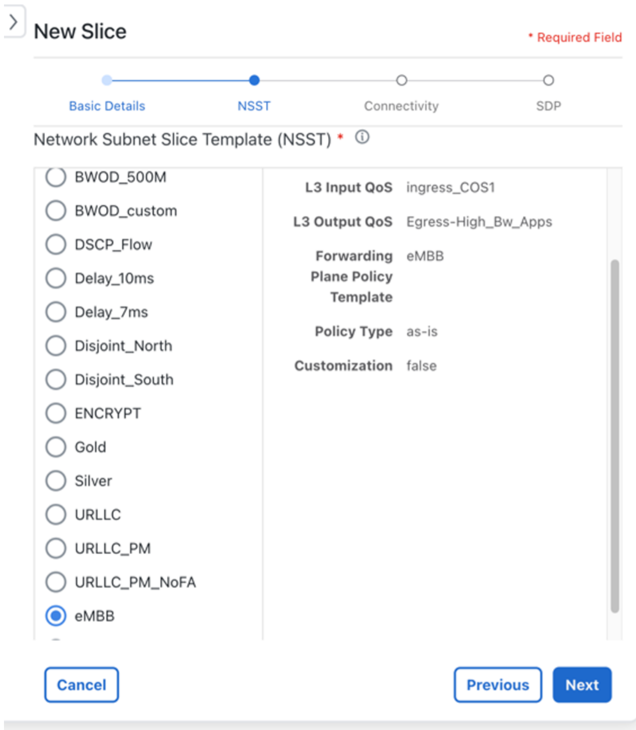



**Step 7** The second step requires Network Subnet Slice Template (NSST) details.

This information specifies which Slice Template to use from the pre-created Template catalog. To match the 3GPP 5G naming convention, it is named the Network Slice Subnet Template (NSST). The description on these templates describes the intent specified by the Slice Designer. Depending on the Slice Service Type selected in the previous step (which was **L3**), the system pulls the appropriate L3 (or L2, if specified) based functionality referenced in the template (for example, QoS settings).

**Step 8** Select the desired intent from the pre-created Slice Catalog: **eMBB**.

*Figure 142: New slice - NSST*




**Note** Slice Designer sets the Slice Catalog names, descriptions, and parameters during the catalog creation phase.

**Step 9** Click **Next**.

**Step 10** The third step requires Connectivity details.

This information builds the connectivity details for the slice by defining whether it is dedicated or shared. If it is a dedicated slice, it can optionally connect to pre-created shared slices (if it is not an L2 P2P slice). Single-sided control will allow for uniform bidirectional policies when connecting to the shared slice (i.e., the dedicated slice policies are used when connecting to shared slice endpoints).

**Step 11** For Connectivity Group, the field will automatically show Default and cannot be changed.

**Note** The IETF Slice YANG model has the concept of Connectivity Groups with the idea that multiple Connectivity Groups can be built under a single Slice ID. Currently, only one Connectivity Group is supported.

**Step 12** Determine the slice Isolation behavior by selecting **Dedicated** or **Shared** for Isolation. In this instance, select **Dedicated**.

**Note** Unique to Crosswork Network Controller, **Dedicated** slices can connect to shared slices (for example, providing a VPN extranet connectivity model).

**Note** If **Shared** is selected, the remaining selections in the Connectivity step default to system details (for example, the Connectivity Type is set to **Any To Any**).

**Step 13** For Connectivity Type, select **Any To Any**.

When selecting the connectivity requirements, choosing L2 or L3 services will determine the available Connectivity Type options.

- L3 Services: **Any To Any**, **Hub, and Spoke**.
- L2 Services: **Any To Any**, **Hub and Spoke**, **Point To Point**.

**Note** If you select **Hub and Spoke**, the endpoint role will be selected at a later step.

**Step 14** In this instance, skip both Connectivity Shared Slices and Bandwidth Reservation.

**Note** Connectivity Shared Slices – If **Dedicated** was selected, the option to connect to an existing shared slice instance becomes available.

**Note** Bandwidth Reservation – If **Allow Customizations** is selected during the catalog entry (and the Policy Type selected is **as-blueprint**), you will have a customizable NSST. You can select Bandwidth Reservation per slice instance or enter a different value.

**Step 15** For Single Sided Control, leave **True** as the default.

**Note** If **True** is selected, it will force a dedicated slice path forwarding behavior towards shared slice endpoints (overriding shared slice path forwarding intent and will ensure the same bi-directional path forwarding).

**Step 16** Select **Show advanced settings** to edit optional parameters.

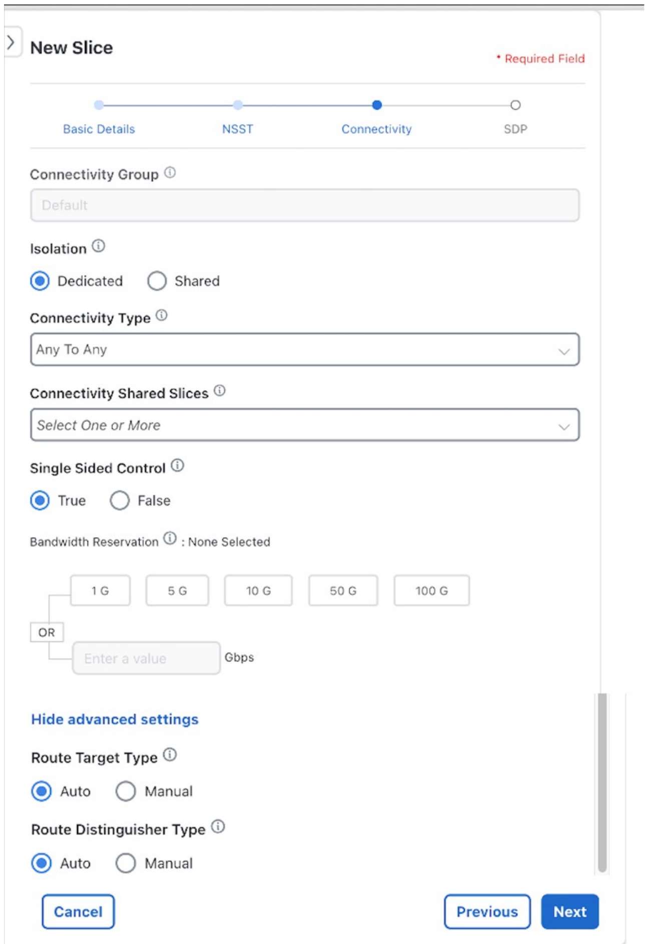
Advanced settings are only available for non-L2 P2P based slices. This will allow for custom Route Target (RT) and Route Distinguisher (RD) settings. By default, these are set to auto and thus do not require configuration. If the connectivity type is any-to-any and manual RT is selected, then a box allowing manual entry of the RT is presented, with this value being used uniformly across all sites to import/export. If the connectivity type is hub-spoke, then two RTs will be required, one for hub and one for spoke.

**Step 17** For Route Target Type, select **Auto**.

**Step 18** For Route Distinguisher Type, select **Auto**.

**Step 19** Click **Next**.

*Figure 143: New slice advanced settings*




**Step 20** The fourth step requires Slice Demarcation Point (SDP) details.

Here, the Slice Requester provides PE endpoint interface details. These endpoints are called SDPs per IETF Slicing standards.

**Step 21** Enter the SDP ID and Attachment Circuit ID to configure the string data. Both entries must be unique within the slice service instance.

- For SDP ID, type **1**.
- For Attachment Circuit ID, type **1**.

**Step 22** In Node ID, select a node from the list: **Node-4 [192.168.255.20]**.

This Node ID uniquely identifies an edge node of the SDP.

**Step 23** Select the Interface Type, **TenGigE**, and type the Interface ID, **0/0/0/10**.

**Step 24** For VLAN ID (optional), type **401**.

**Step 25** For Interface IP, type **172.16..2.1** and type **29** for the mask. This defines the IP address of the attachment circuit.

**Step 26** Since this is an L3 slice service, the following field is required. For the Peering Protocol, select **BGP** as the SDP peering protocol to CE.

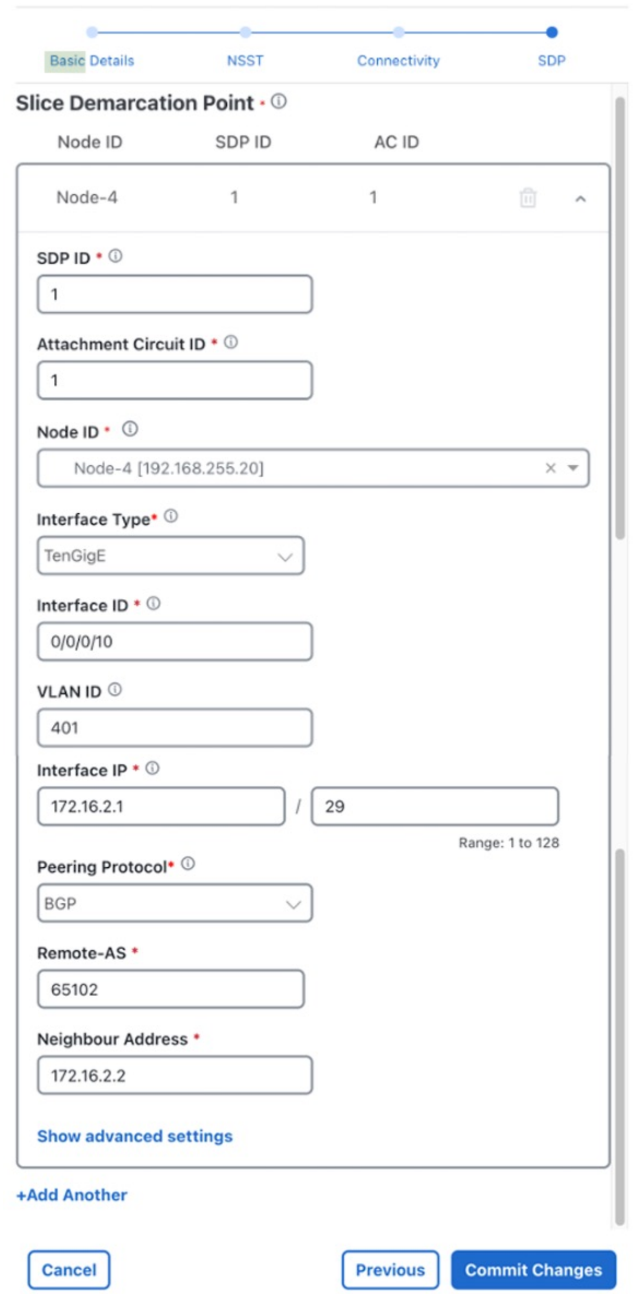
**Step 27** Since the Peering Protocol was defined as BGP, the following fields are required.

**a.** For Remote-AS, type **65102**.

**b.** For Neighbor Address, type **172.16.2.2**.

**Step 28** Click **+ Add Another** to add a second (Node-5) and third (Node-2) SDP endpoints. See the Required Parameter Values table above for parameter values.

*Figure 144: Slice demarcation point*




**Step 29** Click **Commit Changes**.

The new slice service is deployed.

# Step 4 Deploy a slice using NSO CLI (optional method)

The option to deploy a slice using the NSO CLI is also available. The below payload shows the details of deploying a slice using load merge when using the NSO CLI. The defaults are not displayed.

*Figure 145: Deploy slice using NSO CLI optional method*

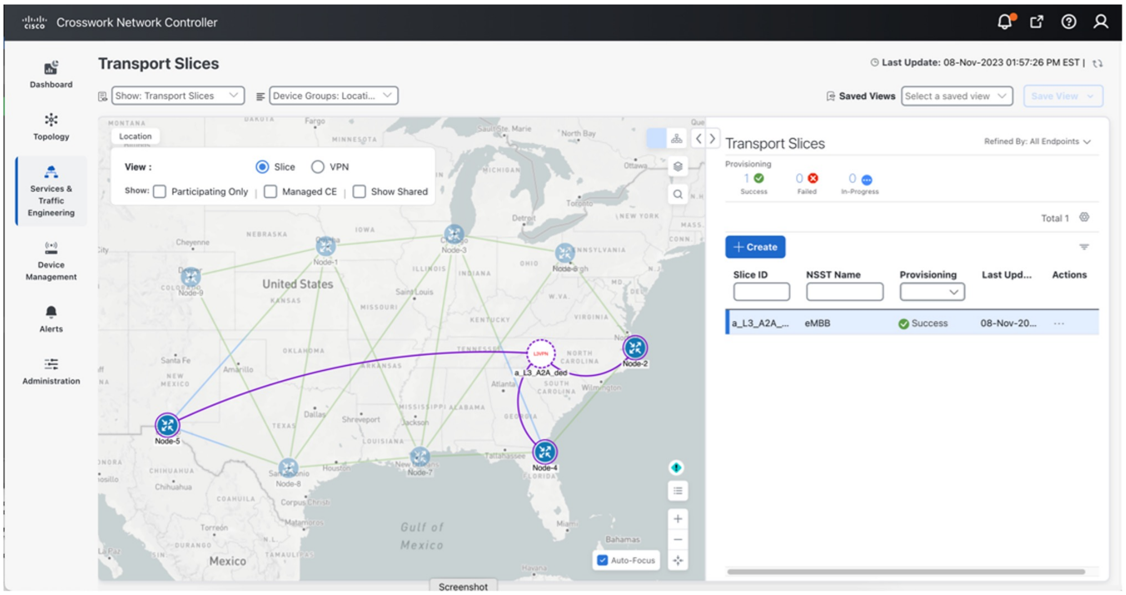
```
network-slice-services slice-service a_L3_A2A_ded
 service-description "L3 any-2-any dedicated slice- a_L3_A2A_dedicated_eMBB.cli"
 service-tags tag-type service-tag-customer
  value [ ACME ]
 !
 service-tags tag-type service-tag-service
  value [ L3 ]
 !
 service-tags tag-opaque nssai
  value [ 123459876 ]
 !
 slo-sle-template    eMBB
 sdps sdp 1
  node-id Node-4
  service-match-criteria match-criterion 1
   target-connection-group-id group1
  !
  attachment-circuits attachment-circuit ac1
   ac-tp-id            TenGigE0/0/0/10
   ac-ip-address       172.16.2.1
   ac-ip-prefix-length 29
   ac-tags ac-tags attachment-circuit-tag-vlan-id
    value [ 401 ]
   !
   sdp-peering protocol peering-protocol-bgp
    bgp-attributes neighbor [ 172.16.2.2 ]
    bgp-attributes remote-as 65102
   !
  !
 !
 sdps sdp 2
  node-id Node-5
  service-match-criteria match-criterion 1
   target-connection-group-id group1
  !
  attachment-circuits attachment-circuit ac2
   ac-tp-id            TenGigE0/0/0/2
   ac-ip-address       172.16.1.1
   ac-ip-prefix-length 29
   ac-tags ac-tags attachment-circuit-tag-vlan-id
    value [ 301 ]
   !
   sdp-peering protocol peering-protocol-bgp
    bgp-attributes neighbor [ 172.16.1.2 ]
    bgp-attributes remote-as 65101
   !
  !
 !
 sdps sdp 3
  node-id Node-2
  service-match-criteria match-criterion 1
   target-connection-group-id group1
  !
  attachment-circuits attachment-circuit ac3
   ac-tp-id            TenGigE0/0/0/2
   ac-ip-address       172.16.3.1
   ac-ip-prefix-length 29
   ac-tags ac-tags attachment-circuit-tag-vlan-id
    value [ 601 ]
   !
   sdp-peering protocol peering-protocol-bgp
    bgp-attributes neighbor [ 172.16.3.2 ]
    bgp-attributes remote-as 65103
   !
  !
 !
 connection-groups connection-group group1
  connectivity-type any-to-any
 !
!
```

# Step 5 Visualize and validate the new slice deployment

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Transport Slices**.

The Transport Slices panel appears, and the new slice is displayed. The Provisioning state should show as **Success**.

***Figure 146: Transport slices - provisioning state***




**Step 2** Optionally, the slice service state can be verified, from the NSO CLI, that all stages were successfully provisioned with all plan states **reached**.

***Figure 147: Slice service state verification from NSO CLI***

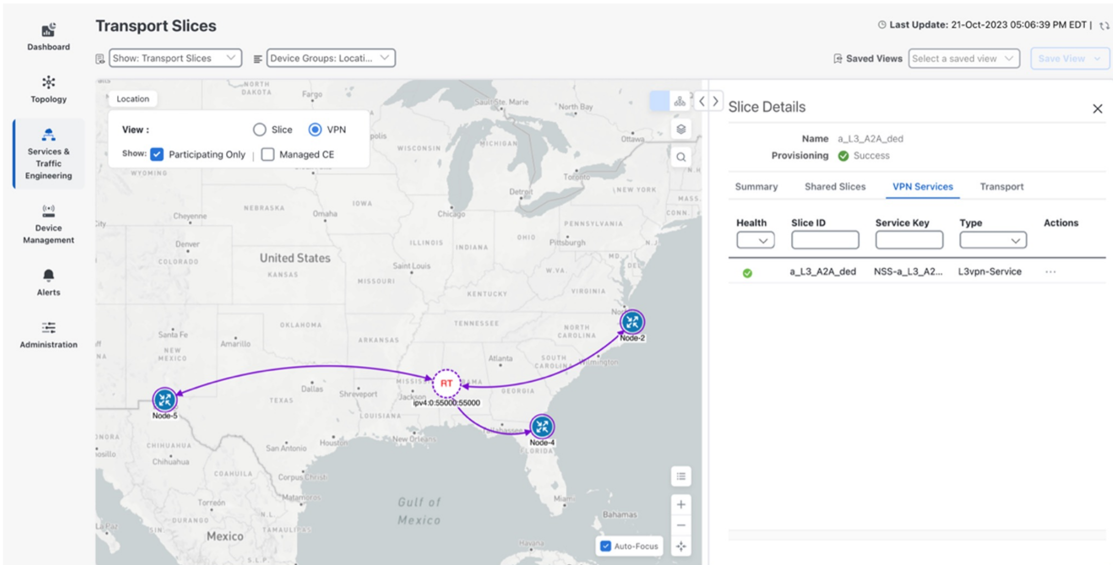
```
admin@ncs# show network-slice-services slice-service-plan a_L3_A2A_ded
                                                                                                   POST
                BACK         STATUS                                                                ACTION
TYPE  NAME  TRACK  GOAL  CODE    NODE    STATE                          STATUS   WHEN                  ref  STATUS
-------------------------------------------------------------------------------------------------------------------
self  self  false  -     -       -       init                           reached  2023-10-21T16:06:21  -    -
                                         ready                          reached  2023-10-21T20:51:31  -    -
sdp   1     false  -     -       Node-4  init                           reached  2023-10-21T16:06:21  -    -
                                         ietf-nss-nano:config-apply     reached  2023-10-21T16:06:21  -    -
                                         ready                          reached  2023-10-21T20:51:31  -    -
sdp   2     false  -     -       Node-5  init                           reached  2023-10-21T16:06:21  -    -
                                         ietf-nss-nano:config-apply     reached  2023-10-21T16:06:21  -    -
                                         ready                          reached  2023-10-21T20:51:31  -    -
sdp   3     false  -     -       Node-2  init                           reached  2023-10-21T16:06:21  -    -
                                         ietf-nss-nano:config-apply     reached  2023-10-21T16:06:21  -    -
                                         ready                          reached  2023-10-21T20:51:31  -    -

 plan status color-allocation-data color 100
 plan status service-tag-service [ L3 ]
 plan status forwarding-plane-policy eMBB
 plan status rt-allocation-data hub-rt 0:55000:55000
```

**Step 3** From the Transport Slices screen in the UI, click ⋯ in the Actions column for the newly created slice, **a_L3_A2A_ded**, and select **View Details**.

The Slice Details panel appears while the topology map updates to show the new slice.

**Step 4** In the Slice Details panel, select the **VPN Services** tab, and in the topology map, select **VPN** as the View. The information updates so you can see that the slice is provisioned with auto-RTs of 55000:55000 and the service is healthy.

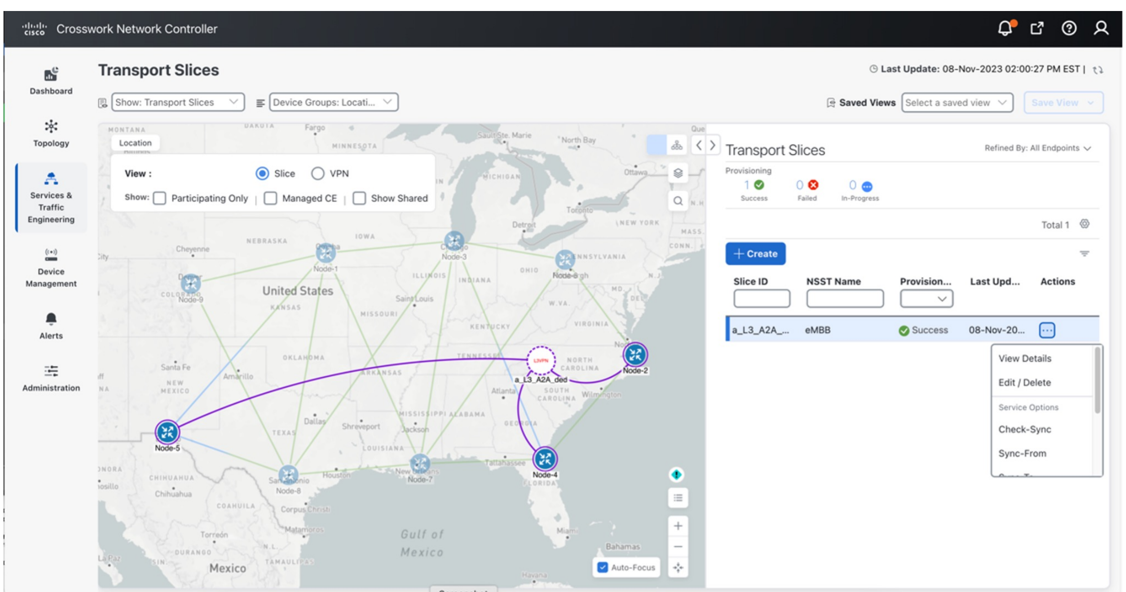
*Figure 148: Slice details panel - VPN services tab*




**Step 5** In the Slice Details panel, click **X** to close the VPN Services tab and Slice Details panel.

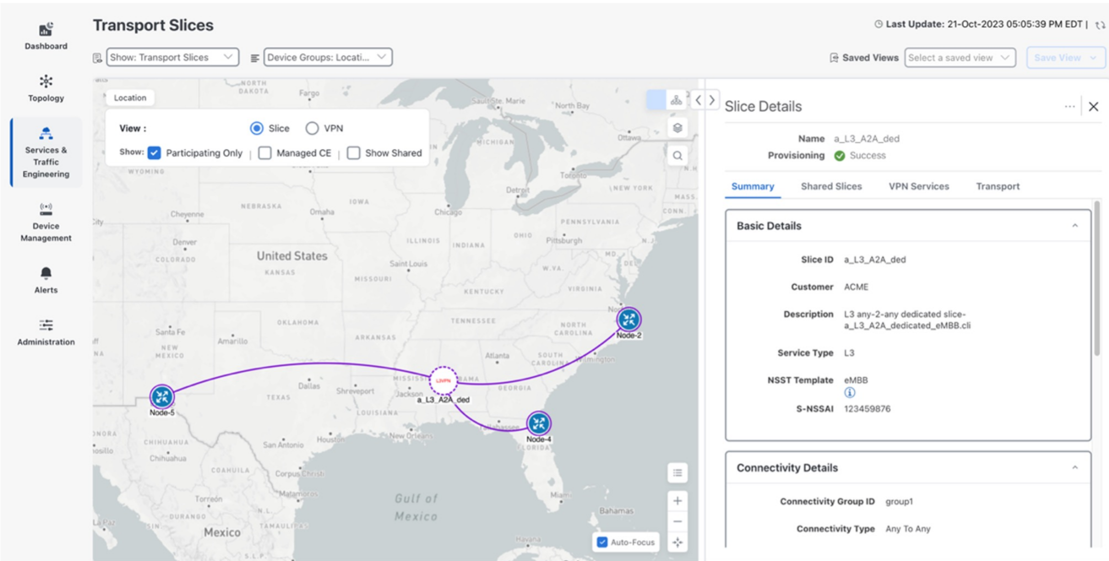
**Step 6** In the Transport Slices panel, click in the Actions column for the newly created slice, **a_L3_A2A_ded**, and select **View Details**.

*Figure 149: Transport slices view details*



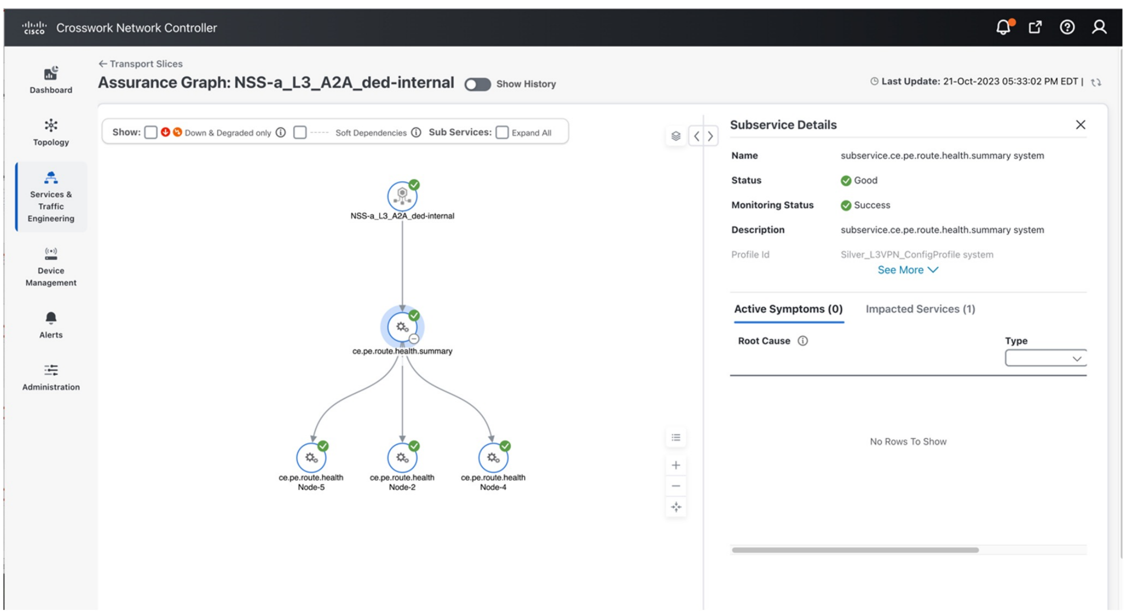

**Step 7** In the Slice Details panel, select the **Summary** tab to view the slice details: Basic Details, Connectivity Details, and Service Demarcation point (SDP).

*Figure 150: Slice details panel summary tab*




**Step 8** Again, select the VPN Services tab. Click ... in the Actions column for the newly created slice, **a_L3_A2A_ded**, and select **Assurance Graph** to view the Monitoring Status and the status of the Service Health components defined in the Service Assurance Heuristics Package that was included in the Slice eMBB intent.

*Figure 151: Assurance graph*

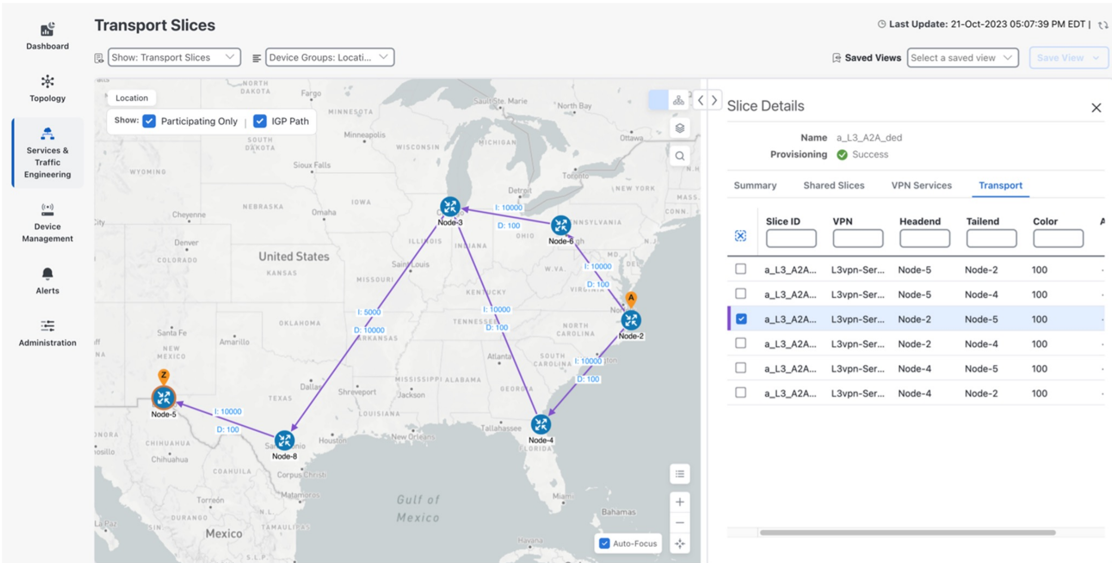



**Step 9** As additional slices are added, you can visualize and validate further details, such as the forwarding path. For example, here is a sample forwarding path for slice traffic from Node-2 to Node-5. Here are a few observations.

- The high BW link between Node-3 and Node-8 (IGP=5k) is used (which was the desired intent), but this link also has a delay of high delay of 10ms (in each direction). Since latency was not an intent objective, this is fine.

- Also, notice that Equal Cost Multi-Pathing is used when available (multiple ECMP paths between Node-2 and Node-3).

*Figure 152: Slice details*




| Slice ID | VPN | Headend | Tailend | Color |
|---|---|---|---|---|
| a_L3_A2A... | L3vpn-Ser... | Node-5 | Node-2 | 100 |
| a_L3_A2A... | L3vpn-Ser... | Node-5 | Node-4 | 100 |
| a_L3_A2A... | L3vpn-Ser... | Node-2 | Node-5 | 100 |
| a_L3_A2A... | L3vpn-Ser... | Node-2 | Node-4 | 100 |
| a_L3_A2A... | L3vpn-Ser... | Node-4 | Node-5 | 100 |
| a_L3_A2A... | L3vpn-Ser... | Node-4 | Node-2 | 100 |

**Step 10** Using the example above, once additional slices are added, if external Accedian probes were also installed at the CPE sites connected to the Slice endpoints at Node-2 and Node-5, an ~11ms latency (each way) between the two sites can be seen (example below). This is accurate because the link between Node-3 and Node-8 has a latency of ~10ms in each direction.

*Figure 153: Accedian Skylight Analytics for external probes between node-2 and node-5*

## Summary and conclusion

As we observed in this example, users can utilize Crosswork Network Controller to create a transport slice that has Layer3 any-to-any connectivity across three endpoints, using the enhanced Mobile Broadband (eMBB) catalog intent. The eMBB intent provides the highest bandwidth available path (including proper QoS marking/scheduling treatment), and some basic service assurance capabilities such as endpoint interface status and PE-CE route health.

# Scenario: Update the slice intent to URLLC_PM

In this scenario, you will update the enhanced Mobile Broadband (eMBB) slice intent, detailed in the previous scenario, to an Ultra-Reliable and Low-Latency Communications with Performance Monitoring (URLLC_PM) slice intent.

*Table 8: Previous eMBB vs. updated URLLC_PM transport slice intent*

| | Previous transport slice intent (Slice using eMBB) | Updated transport slice intent (Updated slice to URLLC_PM) |
|---|---|---|
| Description | Implement a Layer3 any-to-any connectivity using the Enhanced Mobile Broadband (eMBB) slice intent. | Update an eMBB slice intent to an Ultra Reliable Low Latency Communication (URLLC) with Performance Monitoring catalog intent. |
| Features | The eMBB intent provided:<br>• Highest bandwidth available path, including proper QoS marking/scheduling treatment<br>• Basic service assurance capabilities, such as endpoint interface status and PE-CE route health, while also specifying eBGP peer routing connectivity details to CE devices | The URLLC_PM intent will provide:<br>• Lowest latency available path, including proper QoS marking/scheduling treatment<br>• High reliability path<br>• Advanced Performance Monitoring service assurance capabilities, including SR-PM probing |

Before you deploy this update, the Slice Designer must add the URLLC_PM intent into the slice template catalog. As previously discussed, these are one-time operations and can be leveraged for all future slice instances using this intent.

### Assumptions and prerequisites

This scenario assumes the network has already built out the required network capabilities for this intent. However, they will be briefly reviewed here for this scenario. For more detailed explanations, see the Cisco Transport Slice Automation Design Guide.

**Path Forwarding Prerequisites**

The following settings have been pre-configured with the NSO T-SDN SR-TE CFP for the URLLC ODN path-forwarding intent with these properties:

- Use Color 110 to identify the intent.

- PCE is responsible for dynamic path computation.
- The dynamic path computation will be based on Flex-Algo 128 using the latency metric.

```
admin@ncs# show running-config cisco-sr-te-cfp:sr-te odn odn-template URLLC
cisco-sr-te-cfp:sr-te odn odn-templatee URLLC
 color 110
 dynamic pce
 dynamic flex-algo 128
!
```

**QoS prerequisites**

As previously discussed, you (the Slice Designer) has already built various QoS policies in the network devices as a prerequisite.

For this scenario you must:

- create QoS ingress policy “ingress_COS5” which sets the MPLS EXP to 5. This is then used throughout the core network to give this traffic priority scheduling treatment.
- define the egress QoS policy “Egress-LowLatency,” which provides priority scheduling for this traffic.
- ensure to make these QoS policies available to the Slicing Service package.

**Note** Only the new qos-catalog entries are displayed below.

```
admin@ncs# show running-config network-slice-services slo-sle-templates
qos-catalog network-slice-services slo-sle-templates qos-catalog L2 output-qos-policy
Egress-LowLatency description "Low Latency priority egress"
!
network-slice-services slo-sle-templates qos-catalog L2 input-qos-policy
ingress_COS5 description "Treat all as Priority Data"
!
network-slice-services slo-sle-templates qos-catalog L3 output-qos-policy
Egress-LowLatency description "Low Latency priority egress"
!
network-slice-services slo-sle-templates qos-catalog L3 input-qos-policy
ingress_COS5 description "High Priority/Low Latency"
!
```

**Slice service assurance settings**

In this scenario, you will have advanced service assurance requirements that include both passive state monitoring and SR-PM (active probing).

You will use Cisco Crosswork Network Controller Service Health capability and customize the Cisco Crosswork Network Controller system’s pre-built Heuristic packages (ConfigProfiles and Rules), which define the objects to be monitored and the various delay alarming thresholds.

The objective is to always take the lowest latency path, but you can also set the delay threshold that will trigger an alarm (for this example, set it at 25 minutes).

**Note** Customizing the Heuristic packages is outside the scope of this example.

*Table 9: Heuristic profile and rule names for URLLC_PM*

| Custom heuristic profile and rule names used for URLLC_PM | Usage |
|---|---|
| 25ms_L3VPN_ConfigProfile custom<br>Rule-L3VPN-NM custom | L3 profile-name<br>L3 rule-name |
| 25ms_L2VPN_ConfigProfile custom<br>Rule-L2VPN-MP custom | L2 multipoint profile-name<br>L2 multipoint rule-name |
| 25ms_L2VPN_ConfigProfile custom<br>Rule-L2VPN-NM custom | L2 point-to-point profile-name<br>L2 point-to-point rule-name |

In addition to the above Heuristic packages, you must also create:

- Performance Measurement profiles. These profiles will be used on the SR-TE tunnels.
- y.1731 profile for L2 point-to-point slices using the URLLC_PM intent.

The below PM profiles (used for SR-PM) and y-1731 profiles (used for L2 P2P) have been pre-created in the NSO PM service package.

```
admin@ncs# show running-config cisco-pm-fp:pm

pm pm-profiles delay-profile sr-policy profile 2wayDelay
probe computation-interval 60
probe tx-interval    30000
probe protocol       twamp-light
probe measurement-mode one-way
advertisement periodic interval 60
advertisement periodic threshold 20
advertisement periodic minimum-change 1000
!
pm svc-profiles 2wayDelaySvc
performance-measurement delay-profile sr-policy profile 2wayDelay
!
!
```

```
admin@ncs# show running-config l2vpn-ntw y-1731-profile

l2vpn-ntw y-1731-profile Profile-Delay-1
schedule interval 5
schedule duration 5
type delay
probe measurement-interval 60
delay-params statistic delay-two-way
!
delay-params statistic jitter-two-way
!
!
```

# Step 1 Create a slice template catalog entry

The slice designer must add the URLLC_PM intent into the slice template catalog. As previously discussed, these are one-time operations and can be leveraged for all future slice instances using this intent.

Like before, this operation is done once, uses the above components as input, and can cover both L2 and L3 slice instance requests.

This catalog, along with the slice instances themselves, can be built in multiple ways:

- Using Crosswork Network Controller UI
- Using Crosswork Network Controller or NSO Slicing API
- Using NSO CLI, including load merge from a text file

In this scenario, the use of SR-TE Performance Measurement (SR-PM) will require the ODN template to specify modifications to path forwarding using SR-PM commands.

As discussed in the design session, we need to use the "as-blueprint" setting to instruct the slice package to duplicate the referenced ODN template (URLLC_PM) and automatically assign a new color. This will allow the slice automation code to add the new SR-PM configurations.

Since this catalog entry has service assurance requirements, it can only be created using NSO CLI or Crosswork Network Controller API in the initial release. You will find the details in Step 2 Add service assurance into the slice template catalog using the NSO CLI (optional).

```
network-slice-services slo-sle-templates slo-sle-template URLLC_PM
                template-description "Flex-Algo based Low latency Service with Performance

Measurement Delay Probing"
qos-policy L2 input-policy ingress_COS5
qos-policy L2 output-policy Egress-LowLatency
qos-policy L3 input-policy ingress_COS5
qos-policy L3 output-policy Egress-LowLatency
odn forwarding-plane-policy  URLLC
odn forwarding-plane-policy-type as-blueprint
service-assurance heuristics monitoring-state enable
service-assurance heuristics L2 point-to-point profile-name "25ms_L2VPN_ConfigProfile custom"
service-assurance heuristics L2 point-to-point rule-name "Rule-L2VPN-NM custom"
service-assurance heuristics L2 multipoint profile-name "25ms_L2VPN_ConfigProfile custom"
service-assurance heuristics L2 multipoint rule-name "Rule-L2VPN-MP custom"
service-assurance heuristics L3 profile-name "25ms_L3VPN_ConfigProfile custom"
service-assurance heuristics L3 rule-name "Rule-L3VPN-NM custom"
service-assurance ethernet-service-oam md-name foo
service-assurance ethernet-service-oam md-level 4
service-assurance ethernet-service-oam y-1731 id-type icc-based
service-assurance ethernet-service-oam y-1731 message-period 1s
service-assurance ethernet-service-oam y-1731 profile-delay Profile-Delay-1
service-assurance performance-measurement sr-te pm-svc-profile 2wayDelaySvc
service-assurance performance-measurement sr-te delay-measurement profile 2wayDelay
!!
```

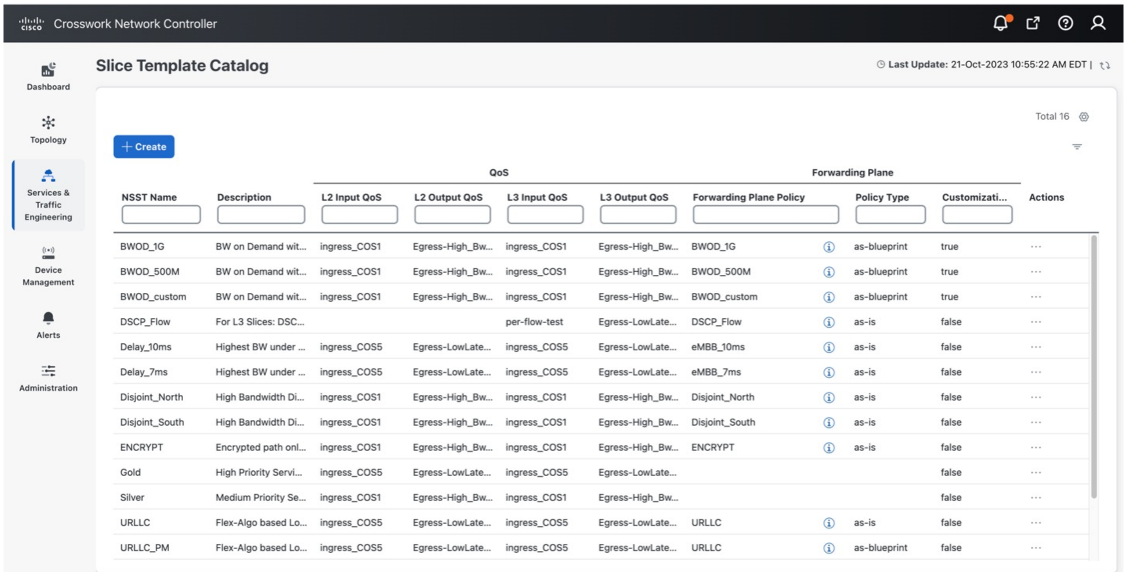
To create a slice template catalog entry using Crosswork Network Controller UI, do the following:

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Slice Template Catalog**.

The Slice Template Catalog screen appears.

**Note** For this scenario, the templates that appear in the image below have already been created on NSO.

*Figure 154: Slice template catalog*




**Step 2** Click **+ Create** to create a new slice catalog entry. The New Slice Template screen appears.

**Step 3** For Network Subnet Slice Template (**NSST**), type the new slice template name: **URLLC_PM**. In addition, in the Description field, type a short description of the slice template's intent: **Low Latency Service**.

**Step 4** Assign the QoS ingress and egress policies. This depends on the slice instance Service Type (L2 or L3 policy), which you will define later when creating a new slice instance.

**Note** For this scenario, the five fields at the bottom of the screen (L2 Input QoS, L2 Output QoS, L3 Input QoS, L3 Output QoS, Forwarding Plane Policy Template) have already been provisioned in NSO and automatically appear as an option in each list.

**Note** You may also refer to the table built earlier and found under QoS in the prerequisites.

- For L2 Input QoS, select **ingress_COS5**
- For L2 Output QoS, select **Egress-LowLatency**
- For L3 Input QoS, select **ingress_COS5**
- For L3 Output QoS, select **Egress-LowLatency**

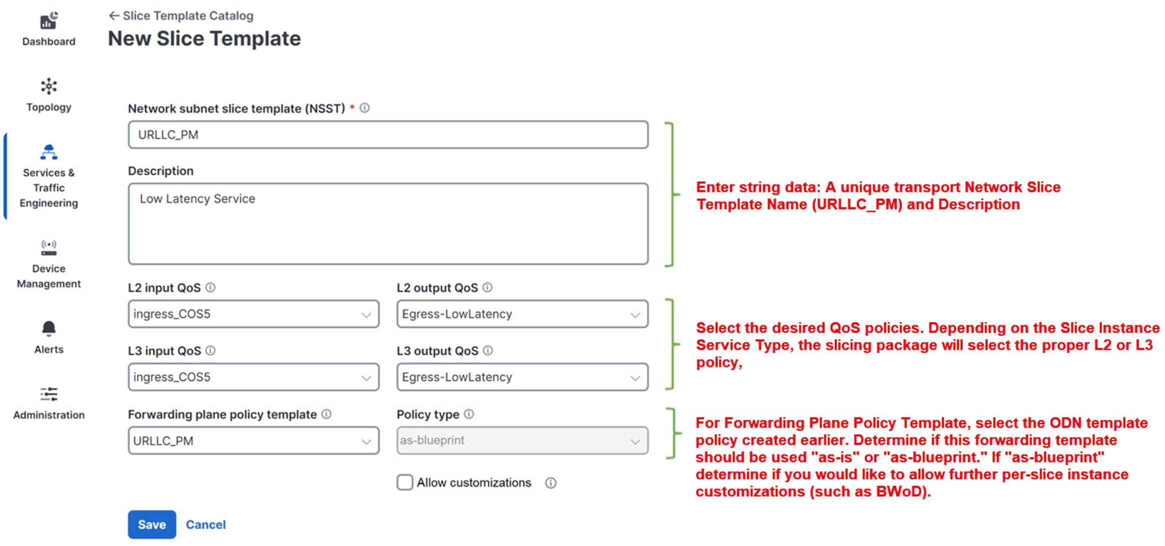
**Step 5** For Forwarding Plane Policy Template, select the ODN template policy created earlier (see prerequisites and assumptions section) that complements the forwarding plane intent. For this scenario, select **URLLC_PM**.

**Step 6** For Policy Type, determine if this template will be used **as-is** or **as-blueprint**. For this scenario, select **as-blueprint**.

When **as-blueprint** is selected, determine if you also want to allow further per-slice instance customizations. These settings determine the SR-TE infrastructure re-use and the scale. Once **as-blueprint** is selected for Policy Type, the **Allow customizations** check box will become available.

**Step 7** Click **Save**.

*Figure 155: New slice template*




# Step 2 Add service assurance into the slice template catalog using the NSO CLI (optional)

If service assurance is required for the slice instance, then the slice designer can add the necessary service assurance functionality into the slice template using NSO CLI or API. Below is a template in NSO CLI with service assurance parameters to be used for this scenario. It can be added directly into NSO CLI or with the API.

There are three service assurance sections in the template settings:

**Step 1** Reference pointers to Crosswork Network Controller Service Health heuristic packages to be used and monitoring state. This monitoring state cannot be changed at the slice instance level at this time; it is set universally for all slice instances referencing this slice template. Various connectivity types (pt-2-pt or multi-point) can be chosen when setting up a slice instance, as well as different service types (L2 or L3). This means there are multiple heuristic package options available, and the system will choose the appropriate package based on the requirements of the slice instance.

**Step 2** If the slice instance is an L2 service type with pt-2-pt connectivity, then Y1731 probe monitoring can be enabled. The required settings are shown in the example below. Slice SLA alarming, and alerting can be configured if the proper settings are selected in the L2 heuristic package for Service Health..

**Step 3** This scenario does not require performance measurement, so it is not included in the template below. However, if desired, it can be enabled in the template, and SR-PM will be dynamically configured on the SR-TE tunnel if the slice forwarding policy-type is set for **as-blueprint**. Slice SLA alarming, and alerting can be configured if the proper settings are selected in the heuristic package for Service Health.

*Figure 156: Slice template catalog*

```
admin@ncs# show running-config network-slice-services slo-sle-templates slo-sle-template URLLC_PM
network-slice-services slo-sle-templates slo-sle-template URLLC_PM
 template-description "Low Latency Service"
 qos-policy L2 input-policy ingress_COS5
 qos-policy L2 output-policy Egress-LowLatency
 qos-policy L3 input-policy ingress_COS5
 qos-policy L3 output-policy Egress-LowLatency
 odn forwarding-plane-policy URLLC_PM
 odn forwarding-plane-policy-type as-blueprint
 service-assurance heuristics monitoring-state enable
 service-assurance heuristics L2 point-to-point profile-name "Silver_L2VPN_ConfigProfile system"
 service-assurance heuristics L2 point-to-point rule-name "Rule-L2VPN-NM-Basic system"
 service-assurance heuristics L2 multipoint profile-name "Silver_L2VPN_ConfigProfile system"
 service-assurance heuristics L2 multipoint rule-name "Rule-L2VPN-MP-Basic system"
 service-assurance heuristics L3 profile-name "Silver_L3VPN_ConfigProfile system"
 service-assurance heuristics L3 rule-name "Rule-L3VPN-NM-Basic system"
 service-assurance ethernet-service-oam md-name foo
 service-assurance ethernet-service-oam md-level 4
 service-assurance ethernet-service-oam y-1731 profile-delay Profile-Delay-1
!
```

See previous explanations from UI figure

CNC Heuristic packages to be used for Service health. Slicing package will pick proper package depending on Slice instance service type (L2 or L3) and L2 connectivity model (point-to-point or multi-point). User can also associate custom packages.

Y1731 probing specifications: For L2 point-to-point slice service types only. Ignored for other slice types

As previously highlighted, slice templates with service assurance parameters can only be created using NSO CLI or Crosswork Network Controller/NSO API at this time. This also means these additional parameters will not be visible when viewing the slice template in Crosswork Network Controller UI.

# Step 3 Update the slice instance using the UI

Once the URLLC_PM slice type catalog has been created, you will use the new template intent data to edit the existing slice instance from the previous scenario and deploy.

For this scenario, use the sample data below:

*Table 10: Required parameter values:*

| Parameter | User value | Mandatory | Notes |
|---|---|---|---|
| slice-service-name | a_L3_A2A_ded | Y | String. Maximum 17 characters. Must be unique. |
| slo-sle-template | URLLC_PM | Y | Selection from pre-built slice catalog |

One of the value propositions of the Cisco slicing solution is that you can use NSO's declarative provisioning capabilities. The system compares the proposed new configuration to the existing configuration and only makes the changes needed to reconcile the differences.

This is called the *minimum-diff* and it is an important capability for declarative systems, as it is not a *rip and replace*. This capability is utilized in this scenario, as the slice template from the previously implemented eMBB L3 any-to-any slice intent is updated to the URLLC_PM slice intent.

In this case, the system will automatically determine the required changes and push them into the devices. These changes will include:

- Re-coloring the VPN prefixes for automated steering over the low-latency-based Flex-Algo 128 path
- Applying new SR-TE policies
- Enabling delay-based Performance Measurements probing

- Telemetry collection
- Monitoring the delay thresholds with new Advanced Heuristic package thresholding.

This can be done either using NSO CLI, Crosswork Network Controller API, or Crosswork Network Controller UI. Below are examples of NSO CLI and Crosswork Network Controller UI approaches.

**Step 1** From the main menu, choose **Services & Traffic Engineering** > **Transport Slices**.

The Transport Slices panel appears.

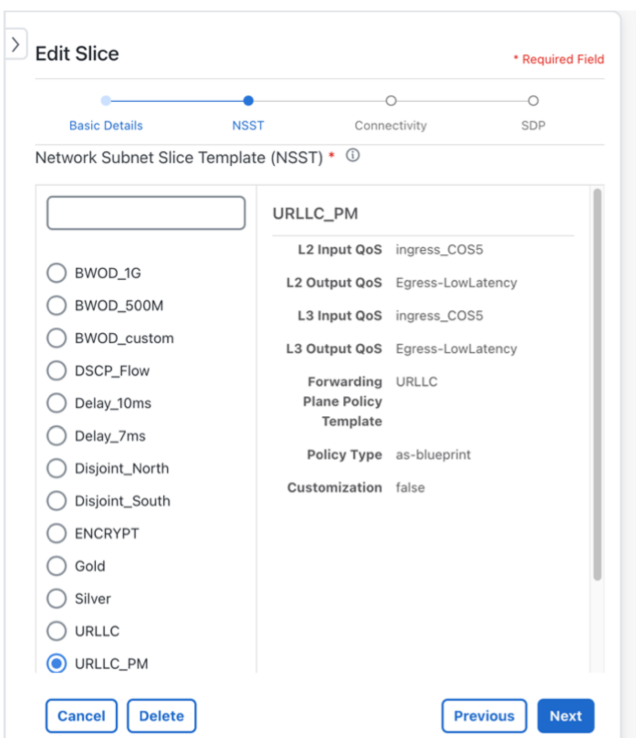
**Step 2** Click in the Actions column and choose **Edit**.

The New Slice panel appears. At the top, four steps are displayed that track the creation of a new slice. The first step requires Basic Details of the new slice.

***Figure 157: New slice steps***


**Step 3** Click **Next** and move to the Network Subnet Slice Template (NSST) selection screen. Select the new intent from the Slice Catalog list: **URRLC_PM**.

***Figure 158: Edit slice***



**Step 4** Click **Next**.

**Step 5** Since this is the only change required, on the Connectivity screen, click **Next** again.

**Step 6** Click **Commit Changes**.

The updated slice service is deployed.

# Step 4 Update the slice instance using the NSO CLI (optional)

The option to deploy a slice using NSO CLI is also available. The payload below shows the details of changing the slice intent using NSO CLI.

```
network-slice-services slice-service a_L3_A2A_ded
service-description "L3 any-2-any dedicated slice- upgrade to URLLC with SRPM"
slo-sle-template    URLLC_PM
!
```

After committing the change, the slice service updates to the new intent.

# Step 5 Visualize and validate the new slice deployment

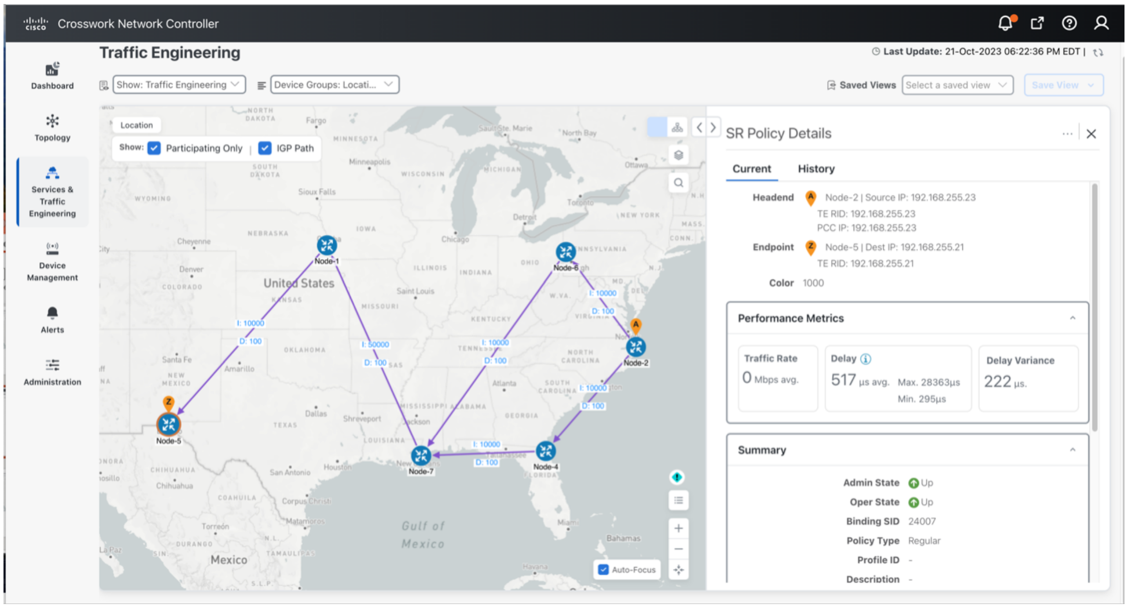
**Step 1** From the main menu, choose **Services & Traffic Engineering > Transport Slices**.

The Transport Slices panel appears, and the new slice is displayed. The Provisioning state should show as **Success**.

**Step 2** If you examine the SR-TE path forwarding between Node-2 and Node-5, you will see the path changed to use the low delay link (yet high IGP metric) between Node-1 and Node 7 to meet the low delay intent.

You can also see that SR-PM measurements have been enabled and both the Delay (517us one-way) and Delay Variance measurements meet the objective.

*Figure 159: SR policy details*




**Step 3** Optionally, you can use testing probes such as Accedian Skylight, and see that the one-way delay has dropped from 11ms to 1ms.

*Figure 160: Accedian Skylight - Cisco Provider Connectivity Assurance*

# Summary and Conclusion

As you observed in this example, users can utilize Cisco Crosswork Network Controller to update an existing transport slice with Layer3 any-to-any connectivity across three endpoints, using the URLLC_PM catalog intent. The URLLC_PM intent provides the lowest latency available path, including proper QoS marking/scheduling treatment, and advanced Performance Monitoring service assurance capabilities, including SR-PM probing.