Cisco Crosswork Network Controller 7.0 Service Health Monitoring

First Published: 2024-05-15

CONTENTS

CHAPTER 1

Introduction

This section explains the following topics:

Audience

This guide is for network administrators who intend to use the Service Health component of the Crosswork Network Controller in their network environment. It assumes familiarity with the following topics:

- Cisco Crosswork Infrastructure and installation of Crosswork applications. For more information, see the *Cisco Crosswork Network Controller Installation Guide*.

- Provisioning L2VPN and L3VPN services

- Networking technologies and protocols (BGP-LS, IGP (OSPF and IS-IS), PCEP, model-driven telemetry)

- Traffic Engineering (TE) tunnels, including

    - RSVP-TE tunnel provisioning

    - Segment Routing Traffic Engineering (SR-TE) policy provisioning

Overview of Service Health

Service Health is a component of Cisco Crosswork Network Controller (Advantage package). For more information on all Crosswork Network Controller solution components, refer to the Cisco Crosswork Network Controller 7.0 Installation Guide.

Service Health extends the capabilities of the Crosswork Network Controller by offering a service-level perspective that goes beyond the physical monitoring of devices and links. While the Crosswork Network Controller can alert you to a failed link or one that has reached its capacity, Service Health assesses how such issues impact the services that traverse these links, such as L2 and L3 VPN services.

Service Health provides operators with the ability to pinpoint exactly where and why a service is degraded, offering targeted tools for service-specific monitoring and assurance.

The application offers APIs and samples for developing additional service monitoring tools as needed. Additionally, it includes ready-to-use packages for monitoring common scenarios, such as:

- Health monitoring of point-to-point L2VPN (for example T-LDP and EVPN) services

- Health monitoring of multipoint L2VPN services (EVPN E-LAN and E-Tree L2VPN EVPN)

- Health monitoring of L3VPN services

- Analysis of health of degraded services to provide information to aid troubleshooting

- Visualization of service health status and its logical dependency tree

Service Health is designed to be extensible, allowing you to add new service monitoring capabilities to meet your requirements.

Service Health Monitoring APIs

Users can integrate other Crosswork applications and third-party applications with service health monitoring by using application programming interfaces (APIs) delivering new capabilities into their network operations.

For more information, see the Cisco Crosswork Network Automation API Documentation on Cisco DevNet.

CHAPTER 2

Get Started

This section explains the following topics:

Before You Begin

We recommend that you familiarize yourself with the following concepts and complete any planning and information-gathering steps:

- Crosswork Network Controller monitors services at two levels - Basic and Advanced.

  - **Basic Monitoring**: This type of monitoring offers the option of monitoring a higher number of services and provides limited sub-service metrics, resulting in lower resource consumption. Additionally, the graphic map renderings are smaller compared to more detailed monitoring.

  - **Advanced Monitoring**: This monitoring approach is supported for lesser number of services as it monitors a larger number of component sub-services and consumes more compute resources, an increased number of sub-service metrics, and larger graphic map renderings.

  For more information, see Service Health Monitoring Scale Information.

- For L2VPN services, Crosswork Network Controller monitors the overall health based on the subservices, while for L3VPN services, the monitoring occurs at the node level.

- Crosswork Network Controller has implemented a rate-limiting process to manage service monitoring requests efficiently. This means that there may be a delay in publishing service monitoring requests if the number of requests raised per minute exceeds a specific threshold. The thresholds are defined as follows:

  - **L2VPN Services**

    - 50 Basic monitoring requests per minute per service

    - 5 Advanced monitoring requests per minute per service

  - **L3VPN Services**:

> • 500 Basic monitoring requests per minute per Vpn-Node
>
> • 50 Advanced monitoring requests per minute per Vpn-Node

The rate-limiting process also extends to the monitoring data. For example, during a restore process, when all Crosswork Data Gateways send data to the Crosswork Tracker component, the rate at which the Crosswork Tracker processes this data and forwards it to Assurance Graph Manager is regulated. This may lead to a delayed reporting of Events of Significance (EOS) following the restore.

An event is triggered with a severity level of wrarning and a corresponding description to notify you of the delay. The event is cleared once Crosswork Network Controller resumes normal publishing of monitoring requests.

• The Crosswork Network Controller can store up to 50 GB of monitoring data. When storage usage reaches 70% of this capacity, it raises an alarm to alert you of potential storage depletion. If more storage is needed, you can configure external storage in the cloud using an Amazon Web Services (AWS) account. See Configure Additional External Storage, on page 49.

• Crosswork Network Controller uses uses a set of rules, expressed in low code format and saved in packages called heuristics packages to monitor the health of the services.

> • A heuristic package contains what to monitor, how to compute the monitored metrics, and symptoms associated with service health degradation. The overall health of the service is determined by applying the rules from the Heuristic Package.
>
> • The default heuristic packages provided with Crosswork Network Controller are referred to as system packages and cannot be altered. Crosswork Network Controller uses these system packages' predefined rules to deploy various testing probes, including Y.1731, TWAMP, SR-PM, and Provider Assurance Connectivity (formerly Accedian Skylight), to evaluate service health and determine whether the service complies with the Service Level Agreement (SLA) (applicable only to Provider Assurance Connectivity probes).
>
> If the default system packages do not fully meet your needs, you have the flexibility to customize them to better suit your specific requirements.You can create a custom heuristic package by exporting an existing package, modifying it, and then importing it back. See About Heuristic Packages, on page 53.

• Extended CLI support using Crosswork Network Controller's system device packages allows for more comprehensive service monitoring capabilities. These packages are capable of deriving exact sensor paths for metric health calculation, and can be installed as a bundle. To add or extend CLI-based KPI collections, you will need support from Cisco Professional Services. Engage with your Cisco account team for more details regarding this.

Getting Started

Service Health is available as part of the Cisco Crosswork Network Controller Advantage Package (refer to the *Get Started* chapter in the Cisco Crosswork Network Controller 7.0 Installation Guide).

You need a functional Crosswork Network Controller environment with devices onboard and services provisioned before you can start monitoring services. The following table includes links to the documents and processes necessary to accomplish those tasks as they are beyond the scope of this document.

Note   To set up and start monitoring services, follow Steps 1 through 6 in the table below. Steps 7 to 9 are optional and cover advanced use cases.

Workflow	Action
1. Install Cisco Crosswork Network Controller Advantage package.	See the Cisco Crosswork Network Controller 7.0 Installation Guide.
2. Do the basic reachability checks from the Crosswork UI.	See Setup Workflow in the Cisco Crosswork Network Controller 7.0 Administration Guide.
3. Create and provision the required L2VPN and L3VPN services.	You can create and provision services using both the Crosswork Network Controller UI or using APIs: • *Orchestrated Service Provisioning* chapter in the Crosswork Network Controller 7.0 Solution Workflow Guide. • Crosswork Network Controller API Documentation on Devnet
4. Determine if you would like to configure additional external storage. **Note** You can configure external storage at any time.	If you anticipate monitoring health of many services, Cisco recommends configuring external storage after you install Crosswork Service Health and before you begin monitoring the services. See Workflow: Manage Stored Data, on page 6.
5. Enable health monitoring for the provisioned services.	Start monitoring VPN services. See Start Service Health Monitoring, on page 11.
6. Establish your operational processes for responding to degraded services.	Deep dive into the impacted services and subservices health, and drill down to the root cause of the service degradation. See Workflow: Analyze the Cause of Service Degradation, on page 7.
7. (Optional) Use SR-PM to probe and monitor links and TE policies in the network.	Use SR-PM to measure performance metrics of TE policies and links. See Workflow: Use SR-PM to Monitor Links and TE policies, on page 8.

Workflow	Action
8. (Optional) Use Cisco Provider Connectivity Assurance (formerly Accedian Skylight) to probe service health.	Using external probes from Cisco Provider Connectivity Assurance can provide additional insights into the health of the service. **Note** Provider Connectivity Assurance integration is available as a limited-availability feature in this release. Engage with your account team for more information. See Workflow: Monitor Service Health using Cisco Provider Connectivity Assurance (formerly Accedian Skylight), on page 8.
9. (Optional) Customize and Import heuristic packages	Service Health offers a default set of heuristic packages for monitoring. If these packages do not fully meet your needs, you have the option to customize these packages to align with your specific requirements.. See Workflow: Customize Heuristic Packages, on page 9.

Service Health Monitoring Workflows

This section outlines the procedures for different scenarios and functionalities identified in the Getting Started, on page 4 section.

- Workflow: Manage Stored Data, on page 6
- Workflow: Analyze the Cause of Service Degradation, on page 7
- Workflow: Use SR-PM to Monitor Links and TE policies, on page 8
- Workflow: Monitor Service Health using Cisco Provider Connectivity Assurance (formerly Accedian Skylight), on page 8
- Workflow: Customize Heuristic Packages, on page 9

Workflow: Manage Stored Data

Crosswork Network Controller provides 50 GB of storage for monitoring data. If that limit is reached, the last recently used monitoring data will be deleted first.

When the storage exceeds 70% capacity, Crosswork Network Controller generates an alarm prompting you to configure external storage in order to save older monitoring data. The actions detailed in the section describe how to monitor storage usage, reduce the amount of data being stored and how to add additional external storage.

Table 1: Workflow: Manage Stored Data

Action	See..
1. Reduce the number of services being monitored by stopping the monitoring for few services. Review the monitoring data that is already stored on your system and delete any data that you no longer need to free up storage space.	Stop Service Health Monitoring, on page 16
2. Switch services that are using Advanced Monitoring to Basic Monitoring to monitor the services in lesser detail.	Edit Existing Monitoring Settings, on page 13
3. If you still need additional storage, configure additional external storage on AWS Cloud.	Configure Additional External Storage, on page 49

Workflow: Analyze the Cause of Service Degradation

This is an operational workflow and it is iterative. Deep dive into the impacted services and sub-services health, and drill down to the root cause of the service degradation in any of the following ways:

Table 2: Analyze the Cause of Service Degradation

Action	See..
1. View monitored services and identify degraded services.	View Monitored Services, on page 31
2. Identify cause of the service degradation.	• Identify Root Causes Using Last 24Hr Metrics, on page 43 • Identify Active Symptoms and Root Causes of a Degraded Service, on page 35 • Identify Root Causes Using Assurance Graph, on page 40
3. Confirm if the reported degradation is a valid issue. In case it is not a valid issue, you may need to adjust the monitoring level (from Basic Monitoring to Advanced Monitoring or vice versa) to ensure accurate reporting of service health. Alternatively, you can modify the system heuristic package to create a custom Heuristic package to resolve the issue of false positive flagging of service health. If the reported issue is valid, proceed to the next step in this workflow.	Edit Existing Monitoring Settings, on page 13 About Heuristic Packages, on page 53

Action	See..
3. Analyze if the service degradation is on account of an issue with device health.	• View the Devices Participating in the Service, on page 46 • View Collection Jobs, on page 47

Workflow: Use SR-PM to Monitor Links and TE policies

To measure the performance metrics of links and TE policies, Crosswork Network Controller can leverage Segment Routing Performance Measurement (SR-PM). When this feature is enabled, Crosswork Network Controller gathers and processes additional metrics such as Delay, Delay Variance, and Liveness to compute the health and determine if any of the metrics have crossed the threshold compliance.

The following workflow describes how to enable SR-PM collection and view performance metrics collected using SR-PM.

Table 3: Workflow to View Performance Metrics collected using SR-PM

Action	See..
1. Enable SR-PM metrics Collection in Crosswork	Enable SR-PM Metrics Collection, on page 17
2. View the performance metrics of the links and TE policies.	TE policies: View Performance Metrics of TE Policies, on page 18 Links: View Performance Metrics of Links, on page 19
3. Ensure that the health of the policy or link is reported accurately without any false issues. If false reporting of degradation is observed, you can create a custom heuristic package by modifying the system heuristic package to provide customized and accurate health reporting.	About Heuristic Packages, on page 53

Workflow:MonitorServiceHealthusingCiscoProviderConnectivityAssurance (formerly Accedian Skylight)

Crosswork Network Controller can use external probing from Cisco Provider Connectivity Assurance (formerly Accedian Skylight) to measure performance metrics of the L3VPN services. These metrics are then compared with the contracted SLA (defined in the heuristic package), and the results are accessible on the UI for further analysis.

Note Monitoring L3VPN services using Provider Connectivity Assurance is supported only with Advanced monitoring and requires a Skylight Essentials license. See Skylight Licensing Tiers for more information.

To add Accedian as a provider in Crosswork Network Controller, follow step 1 and 2 in the table. Follow step 3 to 6 iteratively for operational purposes.

Table 4: Probe and Monitor Service Health using Cisco Provider Connectivity Assurance

Action	See
1. Install the Skylight Solution.	Refer to the Provider Connectivity Assurance Solution documentation and the Skylight installation guide for information on installing the Skylight solution and deploying it with Crosswork Network Controller. **Note** Sign up and create an account with the self sign-up tool to access the Skylight documentation.
2. Add Skylight as a provider in Crosswork Network Controller.	Add Accedian Skylight as a Provider, on page 23
3. Set up Probe sessions for the L3VPN service.	Monitor Service Health using Cisco Provider Connectivity Assurance (formerly Accedian Skylight), on page 22
4. View the metrics in the Crosswork Network Controller UI.	View Probe Session Details, on page 24
5. Analyze the cause of the service degradation.	Identify Active Symptoms and Root Causes of a Degraded Service, on page 35
6. Confirm if the reported degradation is a valid issue. In case it is not a valid issue, you can modify the system heuristic package to create a custom heuristic package for a customized report of service health.	Workflow: Customize Heuristic Packages, on page 9

Workflow: Customize Heuristic Packages

Crosswork Network Controller uses heuristic packages as the core logic to monitor and report the health of services. Heuristic packages define a list of rules, configuration profiles, supported sub-services and associated metrics for every service type. Heuristic packages provided by the system are read-only and cannot be modified.

If you find that the heuristic packages provided by the system do not meet your monitoring requirements, in terms of monitoring metrics or monitoring thresholds, you can create a customized heuristic package that caters to your specific monitoring requirements using the procedures in this workflow.

Note Customizing heuristic packages is not included in the standard Day 2 support responsibilities. For assistance, please reach out to the Cisco account team or contact Cisco Professional Services.

Table 5: Customize Heuristic Packages

Action	See..
1. Analyze your network services. Check the system heuristic packages for rules, sub-services, and metrics to ensure that the system packages do not have the required metrics, services or thresholds already. Determine the package that most closely matches the conditions you wish to identify in your network.	See Basic and Advanced Monitoring Rules, on page 59 and Reference - Supported Subservices, on page 83.
2. Export the package or packages that include the functions you wish to leverage.	About Heuristic Packages, on page 53 .
3. Using the supplied packages as your template build a new package that gathers the data you need to make determinations about the health of the service you want to monitor. In the simplest use case, you may simply need to edit the threshold points based on the SLAs used in your network. In more complicated use cases, you might need to build a Heuristic package from scratch.	Build a Custom Heuristic Package, on page 55
4. Import the customized Heuristic package in Crosswork Network Controller.	Import Custom Heuristic Packages, on page 57
5. Apply the custom package to all the services that should be using it.	Start Service Health Monitoring, on page 11 Edit Existing Monitoring Settings, on page 13
6. Verify that the custom package is providing the monitoring data that you need to meet your requirements.	View Monitored Services, on page 31

Service Health Monitoring Scale Information

You can monitor a maximum of 52000 services in total. This means you can monitor 52000 services with only Basic Monitoring or a combination of Basic and Advanced monitoring not exceeding 52000 services in total with at most 2000 services using Advanced monitoring.

Type of Monitoring	Supports..
Basic Monitoring	52,000 services
Advanced Monitoring	2000 services

CHAPTER 3

Monitor Service Health

This chapter covers the following topics:

Start Service Health Monitoring

Before you begin

The following procedure assumes that you have already provisioned L2VPN/L3VPN services. To create and provision services, refer to the *Orchestrated Service Provisioning* chapter in the *Cisco Crosswork Network Controller 7.0 Solution Workflow Guide*.

To start health monitoring for a service:

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   For a service not currently being monitored as indicated by a gray icon in the **Health** column for which you wish to enable monitoring, click ⊡ in the **Actions**.

Step 3   Click **Start Monitoring**.

Step 4 In the Monitor Service window that appears: .

a) Select the **Monitoring Level** as **Basic Monitoring** or **Advanced Monitoring**.

b) Click a Configuration Profile from the list of profiles that is displayed to select and apply it to monitor the service.

Step 5 Click **Start Monitoring**. The **Health** column of the service gets updated to reflected the health of the service.

Note Once you have started monitoring the health of the service, in the Actions column, click ⬚ to view additional options: Stop Monitoring, Pause Monitoring, Edit Monitoring Settings, and Assurance Graph.

What to do next

If the health of the service is Degraded, identify the root cause for service degradation and take measures to correct the issue. See Analyze Service Health, on page 31 for more information.

Adjust Monitoring Settings

The following topics explain the various monitoring settings you can use to adjust the service health monitoring.

- Edit Existing Monitoring Settings, on page 13

- Pause and Resume Service Health Monitoring, on page 15

- Stop Service Health Monitoring, on page 16

Edit Existing Monitoring Settings

You can adjust the monitoring settings any time after the service health monitoring is enabled. You can update the Monitoring Level for the service from Basic Monitoring to Advanced Monitoring, or from Advanced Monitoring to Basic Monitoring. You can also update the Configuration Profile (from Gold profile to Silver profile or from Silver profile to Gold profile). See About Heuristic Packages, on page 53 for information about Configuration profiles.

To edit the existing monitoring settings:

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 for which you want to edit the monitoring settings.

Step 3   Choose **Edit Monitoring Settings** from the menu.

24-Jul-...   [···]

Edit / Delete

Stop Monitoring

Pause Monitoring

Edit Monitoring Settings

Assurance Graph
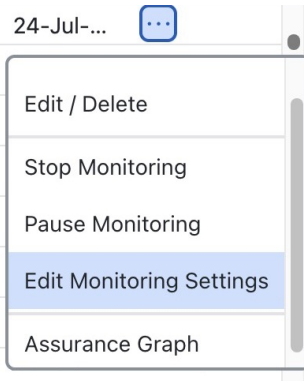
The Edit Monitoring Settings dialog box appears.

Step 4   Choose the **Monitoring Level** or the **Configuration Profile**, as required.

Edit Monitoring Settings

Name	L2-P2P-1381
Monitoring Level	Basic Monitoring ⌄ ⑦

Gold_L2VPN_ConfigProfile system

Silver_L2VPN_ConfigProfile system

Gold_L2VPN_ConfigProfile system

Thresholds to use for Gold L2VPN services

Cpu Threshold Max	70.5 %
Memfree Threshold Min	2000000000 bytes

Cancel   Save

Note   When you switch between Advanced and Basic Monitoring, it can take over 15 minutes for subservice health and active symptoms to become visible.

Step 5   Click **Save**.

A confirmation dialog box appears.

Step 6   Click **Start** *monitoring-type* **Monitoring**.

Crosswork Network Controller starts monitoring the service health using the updated values.

What to do next

If Crosswork Network Controller reports the health as Degraded for the service, identify the root cause for service degradation and take measures to correct the issue. See Analyze Service Health, on page 31 for more information.

Pause and Resume Service Health Monitoring

With this option, you can temporarily pause monitoring the health of services. This is useful when a service is down due to a reported outage or scheduled maintenance, and you don't want to receive notifications about the degradation. If you pause and then resume monitoring, it will continue using the same Basic or Advanced monitoring rules and profile options as before the pause. Additionally, historical data and Events of Significance (EOS) are preserved in the service's history. However, since no data is collected while monitoring is paused, there will be gaps in the historical data for the periods when monitoring was paused.

To pause and resume monitoring the health of the services, 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 that you want to pause the monitoring.

Step 3   Choose **Pause Monitoring** from the menu.

A confirmation dialog box appears. Click **Pause Monitoring**.
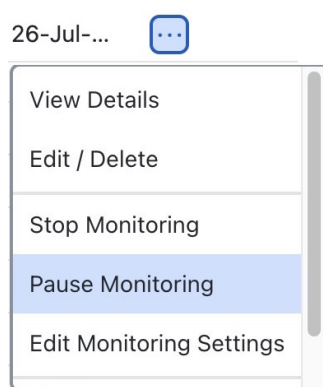
Note   When monitoring is paused, you can still view the Assurance Graph which will show only the top level service with state of paused icon badge and with no child subservices underneath.

Step 4   In the Actions column, if you now click ⋯ for the service that you paused, you will see the **Resume Monitoring** option. Click this option to resume monitoring the service health.

A confirmation dialog box appears. Click **Resume Monitoring**.

When the Crosswork Network Controller resumes monitoring a service after a pause, it utilizes the same monitoring rules and profile options that were in place before the pause.

Stop Service Health Monitoring

When you choose to stop monitoring a service, the system will prompt you to confirm whether you wish to retain the historical monitoring data. The following options are available:

- **Retain Historical Data**: If you choose to retain the historical data, all monitoring information collected prior to the stoppage will remain accessible. This data will be preserved and available for analysis when monitoring is resumed. The monitoring settings will also be retained, ensuring a seamless transition back to active monitoring with historical context.

- **Do Not Retain Historical Data**: If you decide not to retain the historical data, all monitoring settings and historical data will be purged from the database. This action will also delete the Assurance Graph for the stopped service. Subsequent monitoring of the service will start anew, without any reference to previous data.

To stop monitoring the health of a service, 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 stop monitoring.

Step 3   Choose **Stop Monitoring** from the menu.

26-Jul-...   ⋯

View Details

Edit / Delete

Stop Monitoring

Pause Monitoring

Edit Monitoring Settings
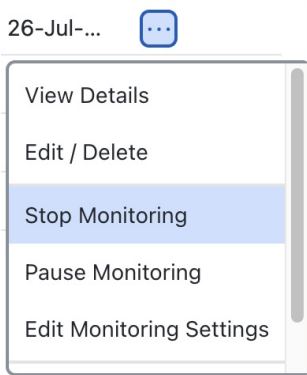
Step 4   The Stop Monitoring dialog box appears. To retain the historical service data for that service, select the **Retain historical Monitoring service for the data** check box.

Stop Monitoring

Name   L3NM-PROBES-3403

⚠   The health of the selected service will no longer be monitored and your monitoring settings will be deleted.
If you want to retain historical monitoring data select the checkbox below.
Are you sure you want to stop monitoring the health of this service?

☑   Retain historical Monitoring service for the data

Cancel   **Stop Monitoring**

Step 5   Click **Stop Monitoring**.

Step 6   If you stopped monitoring a service and selected the **Retain historical Monitoring service for the data** check box, you can start monitoring that same service with historical data still available at a later time. From the **Actions** column of the service, click ⬚ and select **Start Monitoring**.

Enable SR-PM Monitoring for Links and TE Policies

To measure the performance metrics of links and TE policies (SR-MPLS, RSVP-TE), Service Health leverages the Segment Routing Performance Measurement (SR-PM) feature. This feature ennhances the capabilities for troubleshooting and health analysis by providing detailed, historical, and consolidated views of links in the network and transport path metrics. This enables network and service operators to proactively manage, troubleshoot and optimize network infrastructure.

- View Performance Metrics of TE Policies

- View Performance Metrics of Links

Enable SR-PM Metrics Collection

To enable SR-PM metrics collection:

Step 1   From the main menu choose **Administration** > **Settings** > **Data retention** > **Network Performance**. The **Network performance** pane opens on the right.

Step 2   Under **Collect metrics data**, select:

- **LSP PM** - to enable metrics collection for SR policies

- **Link PM** - to enable metrics collection for links
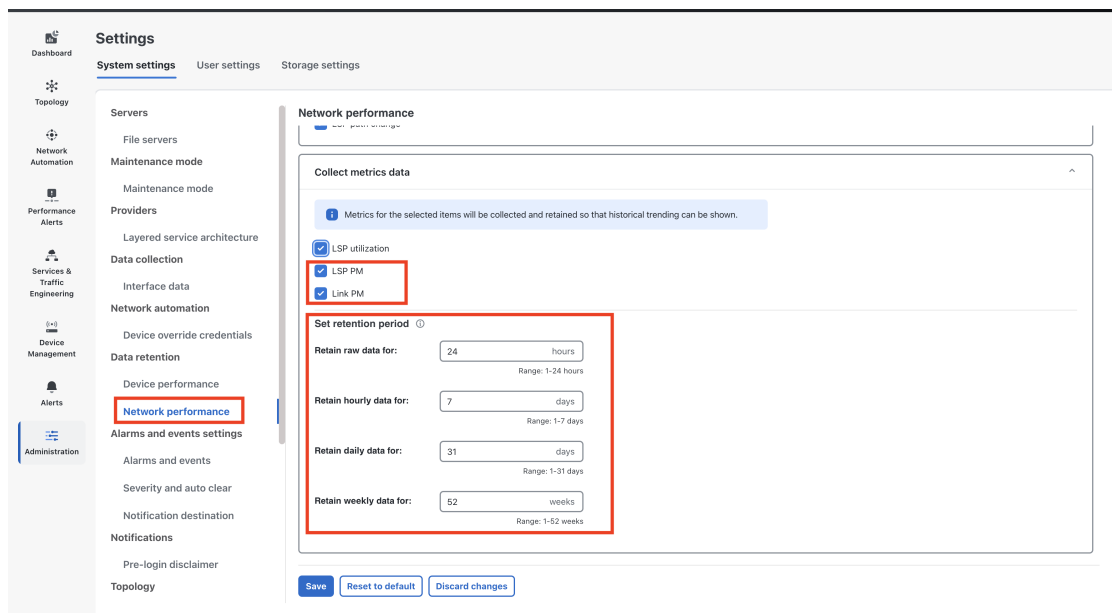
Step 3   (optional) To retain historical data and view trends of these metrics, select the duration for which data should be collected and retained.

Note   Metric data is collected and retained only for the options for which you have enabled SR PM metric collection.

View Performance Metrics of TE Policies

SR-PM data collection is supported for SR-MPLS, SR-CS and RSVP-TE policies. The metric data is used to assess the policy health and indicate if the any of the metrics violated SLAs (which are defined in the Heuristic package). You can view the KPI metrics, as well as the operational and administration status of the service, on the policy tab in the **Service Details** page. If you have enabled data retention, the historical data and trends are available in the **History** tab.

The following metrics are collected for TE policies when SR-PM collection is enabled:

- Delay - available only for SR-MPLS and RSVP-TE policies

- Delay Variance (jitter) - available only for SR-MPLS and RSVP-TE policies

This procedure lists the steps for viewing KPI metrics for a TE policy.

Before you begin

Ensure that you have taken care of the following to view metrics from SR-PM collection:

- Added devices, TE policies and created device groups.

- Enabled SR-PM collection in Crosswork Network Controller and have optionally also enabled data retention to view the historical data and trends.

- Enabled SR-PM metric collection on devices.

✎

Note	Refer to the device-specific documentation for details. These details are beyond the scope of this guide.

d

Step 1   Navigate to the Traffic Engineering topology map. From the main menu, choose **Services & Traffic Engineering** > **Traffic Engineering**.

Step 2   Click the policy tab that you are interested in.

For example, to view policy performance metrics for SR-MPLS policies, click the SR-MPLS tab.

Figure 1: SR-MPLS Policy Performance Metrics in the Traffic Engineering Table
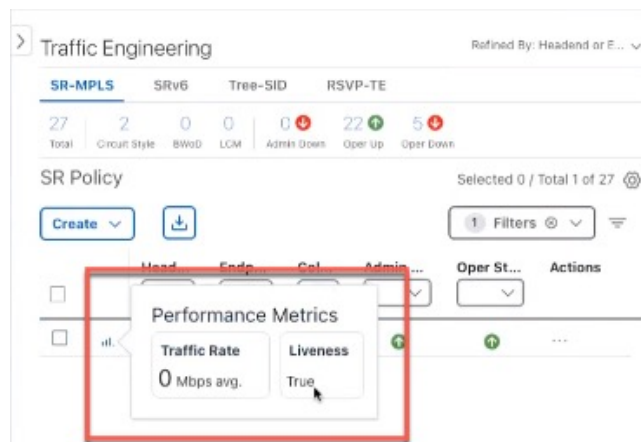
Step 3   Hover your mouse over the graph icon to view the KPI metrics in a carousel view. Alternatively, locate the policy that you are interested in from the TE table. In the **Actions** column, click ⋯ > **View Details**. The **Service Details** page opens and displays the KPI metrics for the policy in the **Performance Metrics** section.

Step 4   To view historical data, click the **History** tab. A chart showing the trends is displayed for each metric here. Click on a time frame in the chart to view the trend of the policy in the selected time.

View Performance Metrics of Links

Link interface metrics are a set of indicators that measure the performance and quality of the communication between two or more network devices. They include parameters such as bandwidth, delay, jitter, packet loss. Link interface metrics can help network administrators to monitor and troubleshoot network issues, optimize network resources, and plan for future network expansion or upgrade.

This procedure lists the steps for viewing link metrics.

Before you begin

Ensure that you have onboarded devices created the required device groups.

Step 1 From the main menu choose **Topology**.

Step 2 Select a link to view its details in any of the following ways:

a) By clicking a link on the topology map

b) By clicking a link from the **Links** tab in the topology map

c) By clicking a link from the **Links** tab in the **Device Details** page.

The **History** tab provides useful insights into the performance and trends of the network. You can select the time interval to analyze the data.

Link details

Summary   History

Name	192_____1 -> 19:_____|2	
State	⬆ Up	
Link type	L3 ISIS IPv4 L2	
ISIS level	2	
Last update	10-Jun-2024 10:03:12 AM IST	

	A side Interface	Z side Interface
Node	**NCS5504-SDN-191**	**NCS55A2-SDN-112**
TE router ID	126.1.1.191	126.1.1.112
IPv6 router ID	2126::191	2126::112
Name	TenGigE0/0/0/9	TenGigE0/0/0/2
Type	ETHERNETCSMACD	ETHERNETCSMACD
IP address	19_____.1	19_____.2
Utilization	0.0058% (584.1Kbps/10Gbps)	0.0086% (864.3Kbps/10Gbps)
In packet drops	0%	0%
In packet errors	0%	0%
IGP metric	10	10
Delay metric	10	10
TE metric	10	10
Admin groups		

Monitor Health of Services using CS-SR Policies

Crossowork Network Controller's support monitoring the health of L2VPN point-to-point services (only IETF:L2VPN:EVPN VPWS) using Circuit-Style Segment Routing (CS-SR) policies.

When the L2VPN service is configured to use circuit-style transport, Crosswork Network Controller automatically initiates monitoring of the service in both directions (A to Z and Z to A) using the `subservice.cssr.policy.health` subservice.

The subservice monitors and reports the **Admin Status**, **Operational Status**, and any flip-flops in the operational status. The operational state of the CS-SR policy is measured using the Liveness metric. This measures if the path is live and capable of carrying traffic, providing a simpler yet effective way to ensure the path's health.

Monitor Service Health using Cisco Provider Connectivity Assurance (formerly Accedian Skylight)

Crosswork Network Controller can leverage external probing, provided by Cisco Provider Connectivity Assurance (formerly Accedian Skylight), to measure metrics of the L3VPN services in the network. The metrics are compared with the contracted SLA (defined in the Heuristic package), and the results are made available on the UI for further analysis.

> **Note**
>
> Monitoring L3VPN services using Cisco Provider Connectivity Assurance is only possible with Advanced monitoring and requires a Skylight Essentials license. See Skylight Licensing Tiers for more information. Sign up and create an account with the self sign-up tool to access the Provider Connectivity Assurance.

High-level Flow

1. When you provision a L3VPN service with probe intent and enable service monitoring, Provider Connectivity Assurance's Orchestrator component learns the probe intent and probe topology from provisioned service.

   The following probe intents are supported:

   - **Agent configurations**: ne-id, VLAN, IP, sub-interface.

   - **Topology**: point-to-point, hub-spoke, full-mesh.

2. Probe sessions with Provider Connectivity Assurance are set up automatically to monitor the service by invoking the relevant RESTConf APIs. The list of RESTConf APIs that are invoked to provision probes sessions are - endpoint, session, service, session activation. The maximum number of probe sessions per service are capped at 200 (for all connection types).

3. Provider Connectivity Assurance's gateway streams the probe metrics to Crosswork Data Gateway, which collects this data using parameterized collection job for Service Health over gNMI. The following probe metrics are collected:

   - Forward and Reverse Delay.

   - Forward and Reverse Variance.

   - Forward and Reverse Packet Loss.

4. The metrics collected during the probe sessions are analyzed and symptoms are raised accordingly, which are then displayed on the Crosswork Network Controller UI.

Add Accedian Skylight as a Provider

Before you begin

Ensure that you have taken care of the following prerequisites before onboarding Accedian Skylight as a provider:

1. Installed the Accedian Skylight software. Refer to the Accedian Skylight documentation for information on installing Accedian Skylight and deploying it with Crosswork Network Controller.

   ✎

   Note   You need an account with Accedian Skylight to access the documentation. Sign up and create an account with the self sign-up tool.

2. Have the following certificates from Accedian Skylight downloaded on your local system or on a folder that can be accessed by Crosswork Network Controller:

   • CA certificate

   • Client certificate

   • Client key

Step 1   **Create a credential profile.**

a) Navigate to **Administration** > **Device Management** > **Credential Profiles** and click + to create a new profile.

b) Enter a name, add the following credential protocols: **HTTPS** and **gNMI**. Add the username and password for both connections.

c) Click **Save**.

Step 2   **Create a certificate profile.**

a) Navigate to **Administration** > **Certificate Management** and click +.

b) Enter a name and select the **Certificate Role** as **Accedian Provider Mutual Auth**

c) Upload the certificates (ca_cert.pem, client_cert.pem, and client_key.key).

d) (Optional) Enter the passphrase for the certificate chain.

e) Click **Save**.

Step 3   **Add Accedian Skylight as a provider in Crosswork Network Controller.**

a) Navigate to **Administration** > **Manage Provider Access**.

b) Click + and enter details in the fields as follows:

   • **Provider Name**: Enter a name.

   • **Credential profile**: Select the credential profile that you created for Accedian Skylight.

   • **Family**: Select ACCEDIAN_PROXY.

   • **Certificate profile** : Select the Accedian Skylight certificate profile.

   Note   This field is displayed after you select the **Family** as ACCEDIAN_PROXY.

   • **Connection types**: Supported protocols are automatically updated from the Accedian credential profile.

• **IP addresses**: Enter the IP address or the Fully Qualified Domain Name (FQDN).

Important If the server certificates present in Accedian Skylight are generated using a Fully Qualified Domain Name (FQDN), enter the FQDN only in this field. Do not enter an IP address. Entering an IP address when the server certificates are generated with FQDN will cause issues in Accedian provider authentication and reachability.

• **Ports**: Enter 443 for HTTPS and a port value for GNMI.

• **Encoding Type**: Select PROTO.

Note   Only encoding of type **PROTO** is supported.

c) Click **Save**.

What to do next

Confirm that the Accedian Skylight provider is reachable from Crosswork Network Controller. See Check Accedian Skylight Reachability, on page 24.

Check Accedian Skylight Reachability

To check reachability of Accedian Skylight Provider:

1. Navigate to **Administration** > **Manage Provider Access** from the main menu.

2. Ensure the Accedian Skylight provider shows a green reachability status without errors.

Note   If there are certificate errors, the provider will be displayed as **Degraded** and not reachable.

3. The Accedian Skylight provider might still be displayed as reachable on the Crosswork Network Controller Providers list page in spite of the following issues:

   • Invalid HTTPS credentials.

   • Incorrect ports, IP addresses, or credentials for the gNMI protocol (since reachability checks for gNMI are not performed).

4. After resolving any certificate or HTTPS credential issues, delete and onboard the Accedian Skylight provider in Crosswork Network Controller again.

View Probe Session Details

Details from Provider Connectivity Assurance probe sessions for L3VPN services and Y1731 probe sessions for L2VPN services are displayed separately in the **Probe Sessions** tab of the service.

To view the metrics from a probe session:

Step 1 Go to **Services & Traffic Engineering > VPN Services**. The map opens on the left side of the screen and the table opens on the right side of the screen.

Step 2 For the service you are interested in, in the **Actions** column, click **View Details**.

Step 3 In the **Service Details** page that is displayed, click the **Probe Sessions** tab.

> Service Details   ··· | ✕
>
> | | |
> |---:|:---|
> | **Name** | EP45-L3NM-IGP-405-Probe |
> | **Provisioning** | ✔ Success |
> | **Health** | Degraded |
> | **Monitoring Status** | ✖ Error |
> | **Monitoring Settings** | Advanced \| Gold_L3VPN_ConfigProfile custom ⓘ |
>
> Health   Transport   Configuration   ⚓ Path Query
>
> Active Symptoms (8)   **Probe Sessions (3)**
>
> Reactivate Probe   Filter 0 / Total 3   ⚙   ☰
>
> | Health | Probe ... | A Devi... | A Inter... | Z Device | Z Inter... | Actions |
> |---|---|---|---|---|---|---|
> | �旭 ✔ | ✔ | CL4-PE... | Gigabit... | CL4-PE... | Gigabit... | ··· |
> | �旭 ✔ | ✔ | CL4-PE... | Gigabit... | CL4-PE... | Gigabit... | ··· |
> | �旭 ✔ | ✔ | CL4-PE... | Gigabit... | CL4-PE... | Gigabit... | ··· |

Step 4 Click the graph icon next to a probe session for a detailed view of the performance metrics.

If a metric has crossed the defined threshold, a red icon is displayed in the corresponding performance metrics dashlet.

Step 5   To view the Performance Metrics for a service in a carousel view, click the ⬚ icon in the **Actions** column.

The **Probe Session Details** window opens displaying the metrics in a carousel view.

Note   If there are any probe provisioning errors, the monitoring status of the service is **Monitoring Error**. Click the **Reactivate Probe** to restart the probe session for the service. If the probe session reactivates successfully, the **Probe Sessions** page automatically updates with the new metrics.

Probe Session Details

Details   Historical Data

Performance Metrics

Probe Delay Forward ✓	Probe Delay Reverse ✓	Probe Variance Forward ✓
8.601 usec avg	1.560 usec avg	10.361 usec avg
Thresh.10000.000 usec	Thresh.10000.000 usec	Thresh.2000.000 usec

Summary

Service Name	EP45-L3NM-IGP-405-Probe
ProbeSession ID	646c19e1-758a-529c-a870-6d3e39122355
Subservice ID	ss-f6248e84-3205-480f-b251-5f1d111f8f4d
A Device	A CL4-PE-A
A Interface	GigabitEthernet0/0/0/0
Z Device	Z CL4-PE-C
z Interface	GigabitEthernet0/0/0/0

The **History Data** tab provides probe metrics data ranging from the past 90 days up to the most recent 24 hours. See for more information.

What to do next

- If you find that a service is degraded, analyze the root cause of the degradation to troubleshoot the health of a degraded service. See for more information.

View Historical Data from Probe Sessions

To view historical data from a probe session, click the **History data** tab in the **Probe session details** page. This tab displays data from the time monitoring was enabled for the service.

Note   If monitoring was stopped and started again later, the **History data** tab will display data only from the time monitoring was restarted. If you chose to retain historical data while monitoring was stopped, that data is preserved and appears in the **Show History** tab of the Assurance Graph of the service and not in this tab.

Charts displaying aggregated average metric data along with their timestamps are shown. To view data from a specific range, select the desired range from the dropdown menu.
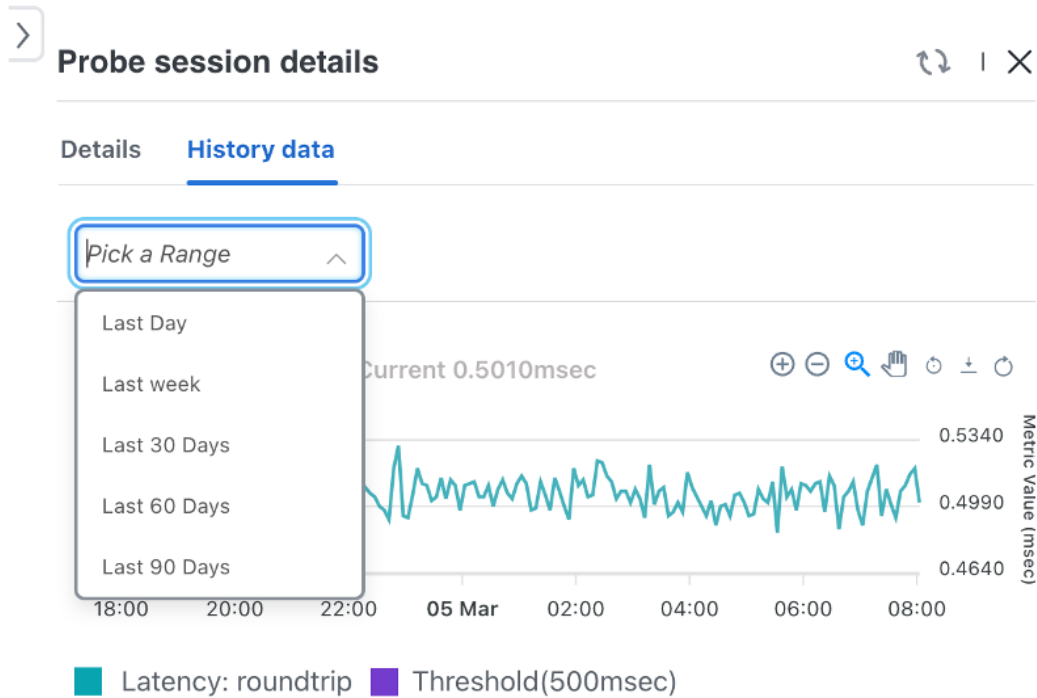
Note   There may be a difference in the first timestamp displayed in the historical chart metric data. This is because, the timestamp displayed is in the operational time zone, while the aggregated timestamp is in UTC format with 00 hours.

Historical Probe Metrics Period	Interval	Aggregate interval
1 week (7 days)	2 hours	2 hours
1 month (30 days)	1 day	1 day (00 hours, start of the day in UTC)
2 months (60 days)	3 days	1 day (00 hours, start of the day in UTC)
3 months (90 days)	1 week (7 days)	1 day (00 hours, start of the day in UTC)

Known Issues and Limitations with Provider Connectivity Assurance

The following is a list of known issues and limitations when Cisco Provider Connectivity Assurance(formerly Accedian Skylight) is deployed for probing service health:

1. Entering an IP address instead of the FQDN when server certificates are generated with FQDN will cause issues with provider authentication and reachability. In this case, the Accedian provider is shown as **Degraded** in the Crosswork Network Controller Providers list page (**Administration** > **Manage Provider Access**).

2. The Accedian Skylight provider is shown as reachable always in the Crosswork Network Controller Providers list page in spite of the following issues in the Accedian Skylight provider credentials:

   • Invalid HTTPS credentials

   • Incorrect ports or IP addresses or credentials for gNMI since there are no reachability checks for gNMI

   In these cases, services monitored by Accedian Skylight probes will have the health as **Degraded** with the symptom as '*Accedian provider does not exist in DLM*'. The symptoms are not cleared until you add the Accedian Skylight provider again, pause and resume the service monitoring.

3. When monitoring is enabled for a service with probe intent but the Accedian Skylight provider is not added in Crosswork Network Controller, an error about the provider not being available is displayed for each of the probe metrics associated with the subservice.

4. You cannot delete the Accedian Skylight provider when a probe session is active.

5. The **Active Symptoms** tab displays the observed value of the metric at the time the symptom occurred, while the **Probe Sessions** tab is constantly updated with the live values of the metrics. Therefore, check the **Probe Sessions** tab for the real-time values of the performance metrics.

C H A P T E R **4**

Analyze Service Health

This section explains how service health monitoring helps in analyzing the health of a service using the metrics displayed in the UI. It guides you on investigating degraded services and subservices to identify the root cause of service degradation.

View Monitored Services

You can view the monitored services in any of the following ways:

From the Crosswork Home Page

Figure 2: VPN Service Health Dashlet
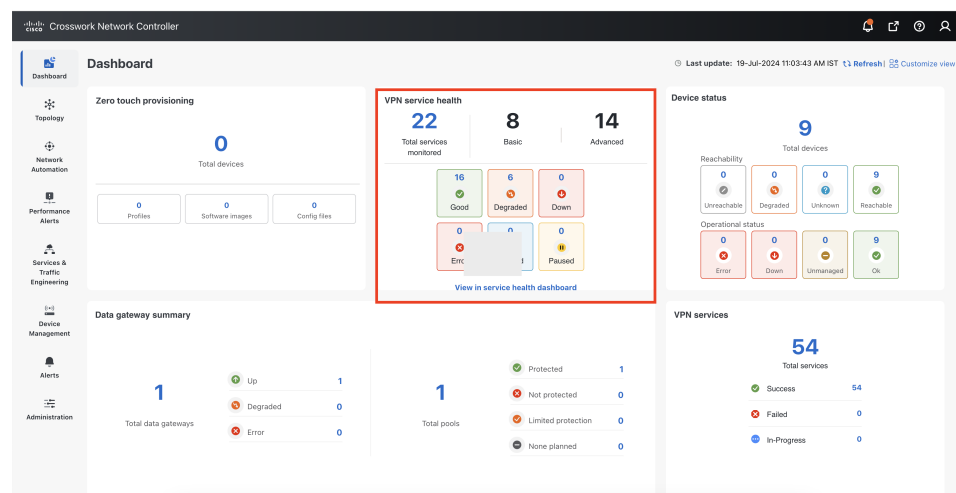
You will see the **VPN Service Health** dashlet on the Crosswork Home Page. This dashlet provides an overview of all the VPN services that are being monitored. From the dashlet you can click any of the status indicators to be taken to the **VPN Services** page with a filter set for the status you selected. To view the degraded services, click the **Degraded** box within the dashlet. This will take you to the VPN Services page, where only the degraded VPN services are displayed.

From the VPN Services Page

From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. All the VPN services are listed on this page. The degraded services show an orange icon in the **Health** column.

You can filter the services by their health (Down, Degraded, Good, Paused, Initiated, Error, Unmonitored). You can also click the Degraded tab in the Health tab in this page to filter and view only the Degraded services.

To clear the filter, click **X** next to the designated filter appearing in the space at the top of the column. To remove all the filters and to show all the VPN services, click the **X** icon in the Filters field above the table.

Note If a service is not yet being monitored, a gray icon is displayed in the Health column. To enable monitoring for such a service, click [...] and select **Start Monitoring**. For more information, see Start Service Health Monitoring, on page 11.

Use the Service Health Monitoring Dashboard

To access the dashboard, click **View in service health dashboard** in the **VPN Service Health** dashlet on the Crosswork Home Page. The **Service Health Dashboard** displays consolidated view of L2VPN and L3VPN services including the total number of provisioned Services and number of monitored services. The dashboard also displays active monitoring sessions for L2VPN and L3VPN services, and indicates any SLA breaches for measured metrics such as latency, jitter, and packet loss in both directions.

Figure 3: Service Health Dashboard
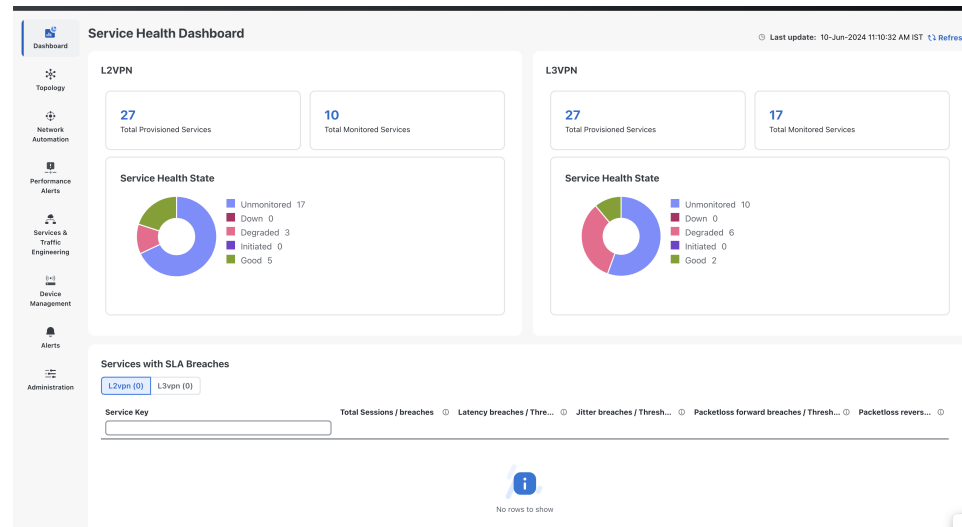
Clicking on any of the status indicators redirects you to the **VPN Services** page with a filter set for the status you selected.

View Monitoring Status of a Service

You can view the **Monitoring Status** of a service from its **Service Details** page.

From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. Locate the service that you are interested in and under the **Actions** column, click **View Details**. This page displays the **Monitoring Status** and the **Health** status of the service.

Monitoring status for a service can be either **Healthy** or **Error**.

- **Healthy**: This means the end-to-end flow of monitoring the service is working as expected and Crosswork Network Controller is able to evaluate the health of the service successfully.

- **Error**: This means Crosswork Network Controller is unable to monitor the current health of the service due to component failures, operational errors or device errors, and the health status that is displayed is the last known health of the service. You can filter monitoring errors using the mini dashboard or the filters.

> **Note** Monitoring errors reported on account of device health do not affect the overall health of the service.

In the Historical Graph, Events of Significance (EoS) are displayed for monitoring errors as well. If the service is healthy but there are monitoring errors, a green warning icon is displayed. However, if the service is degraded and there are monitoring errors, then an orange warning icon is displayed. Clicking these icons provides you with the details in the symptoms table with type as **Monitoring Errors**.

Note   The Historical Graph displays monitoring errors only when the Monitoring Errors setting is enabled via API. There is no option to enable this setting from the UI. Once this setting is enabled, the system starts to log these monitoring errors as Events of Significance and display them in the historical graph. Refer to the API documentation on Cisco Devnet for more information.

Identify Active Symptoms and Root Causes of a Degraded Service

By analysing 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.

Note   L3VPN service monitoring is supported on Cisco IOS XR devices and not on Cisco IOS XE devices. For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status will remain in the degraded state with a monitoring status as Monitoring error. Stop and restart the service monitoring to recover from this error.

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

Step 1   From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The service assurance dependency graph 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).

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

Service Details   ···   ✕

Name	P2P-SR-C-101
Provisioning	✅ Success
Health	🔻 Degraded
Monitoring Status	❌ Error
Monitoring Settings	Advanced | Gold_L2VPN_ConfigProfile system ⓘ

Health   Transport   Configuration   ⤳ Path Query

✕

Symptom Details ⌄

Name	PCEP Session Health degraded. Device: CL2-PE-C, PCC-Peer: 192.168.15.42
Sub Service	subservice.pcep.session.health system
Last Updated	28-Jul-2023 11:29:20 PM IST

Failed Subexpressions & Metrics ⌄

Show Only Failed ⬤○   Expand All \| Collapse All

Name	Expre
explabel	pcc_p
🔻 pcc_peer_state == 'up'	false

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.

Related Information

- To monitor the VPN services using Assurance Graph capabilities and inspect any services or related nodes that are degraded, see Identify Root Causes Using Assurance Graph, on page 40.

- To identify the issues with the degraded services within a specific time range, use the Last 24Hr Metrics. For details, see Identify Root Causes Using Last 24Hr Metrics, on page 43.

- To identify a service health issue by examining the collection jobs, see View Collection Jobs, on page 47.

About Assurance Graph

In Crosswork Network Controller, a service instance comprises various subservices, each assured independently. The overall health of the service depends on the health of these subservices. The Assurance Graph visually represents the service instances and their dependent subservices in a graphical format. The topmost node in this logical dependency tree represents the monitored service instance, while the child nodes represent its subservices, which may further depend on other subservices.

This graphical representation helps locate problem areas and provides indications of possible symptoms and impacting metrics, aiding in troubleshooting degradation issues. Crosswork Network Controller updates the Assurance Graph automatically when the service instance is modified.

Note   For L3VPN services, Crosswork Network Controller monitors service at the node level. See Assurance Graph for L3VPN Services, on page 38 for more information.

To view a service in the Assurance Graph, from the **Actions** column for the service, select **Assurance Graph**. The Assurance Graph displays the graph on the left pane and details of the service on the right pane. Toggle the **Show History** button to view the historical data chart. Each dot on the history chart represents one Event Of Significance (EOS) for a service.
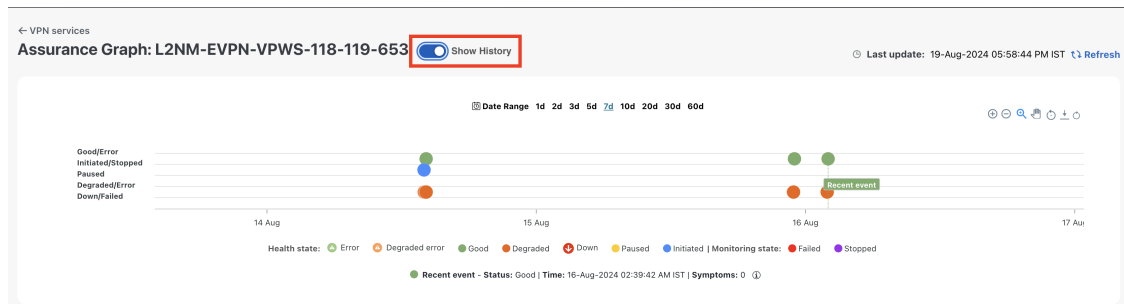
For each EOS, you can view the Assurance Graph and symptoms with 24 hours of metrics collected based on the EOS time. For example, for a service for which monitoring was stopped, a dot appears indicating that the monitoring was stopped. Clicking and dragging over a selected range on the EOS allows you to zoom in on the time range, and hovering over the EoS provides details about the event and associated symptoms.

Assurance Graph for L3VPN Services

For L3VPN services, Crosswork Network Controller monitors the service and the builds the Assurance graph at the node level. The graph includes a summary node for each device and feature-level nodes under each summary node. Nodes with dependencies spanning other nodes (for example `path.sla.summary`) have a feature-level summary node in the graph.

Select endpoints

The Assurance Graph builds its view based on the data-sending endpoints (headends) of VPN nodes. If the Assurance graph becomes too cluttered with more than 50 nodes, Crosswork Network Controller indicates that the graph is too large to display. Use the **Select endpoints** option above the graph to view up to 50 endpoints at a time.

The Assurance Graph filtering is based only on the VPN nodes and does not support filtering by a combination of VPN nodes and endpoints. For example, in a service with 2 VPN nodes, each having 2 endpoints (totaling 4 endpoints), deselecting one endpoint using the **Select endpoints** option will not update the Assurance Graph. The graph updates only when both the endpoints for a VPN node are removed, leading to the entire VPN node being removed from the Assurance graph.

Note For a service, the **Transport Tab** displays all discovered transports related to selected VPN nodes, considering both headend and tailend roles based on the import/export policy configured in the service intent. When you use the **Select endpoints** option and deselect a headend endpoint, the **Transport** tab updates to remove the headend endpoint from view but may still show the tailend endpoint if it is relevant to other headends.

In contrast, the **Assurance Graph** focuses only on headend endpoints. If you deselect a endpoint and no other endpoints are left for that node, the graph removes the entire node from the display.

Show History

When you toggle the **Show History** button to view the historical data chart, you'll see two types of events: **VPN Node events** and **Global events**. The event type is indicated in the description of the Event of Service (EoS) when you hover over it.

- **Global Events**: These events span multiple VPN nodes. For example, an EoS in the probe service (`path.sla.summary`) is classified as a Global event.

- **VPN Node Events**: These events are specific to a single VPN node.

In **L3VPN services**, symptom counts are shown at the VPN node level, with the Device ID (VPN node name) displayed alongside the symptoms. The timeline series in the Show History view displays these symptoms at the VPN node level (endpoint).
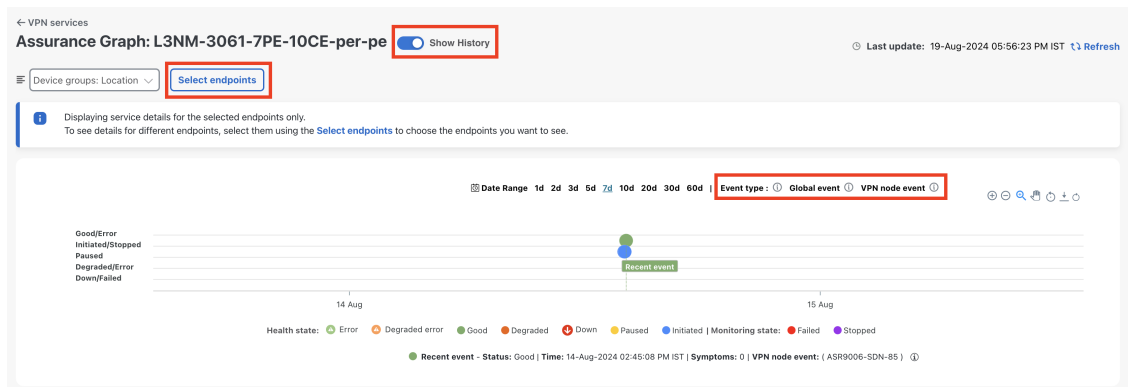
The **Service Details** will continue to show the total symptoms count of the service, for the selected EoS time.

Note If endpoints are selected, the total symptom count indicates the total symptom count of the selected endpoints.

Service details

In the **Service details** page, the **Active Symptoms** tab shows the health details of feature-level nodes, including the number of subservices in the **Up** or **Degraded** state. Clicking on the **Degraded** state in a feature node, filters the table to display symptoms and monitoring errors only for that node.

Identify Root Causes Using Assurance Graph

You can use the Assurance Graph to inspect and drill down to the root cause of a service degradation.

Before you begin

Ensure that service health monitoring is enabled for the service you want to inspect. For details, see Start Service Health Monitoring, on page 11.
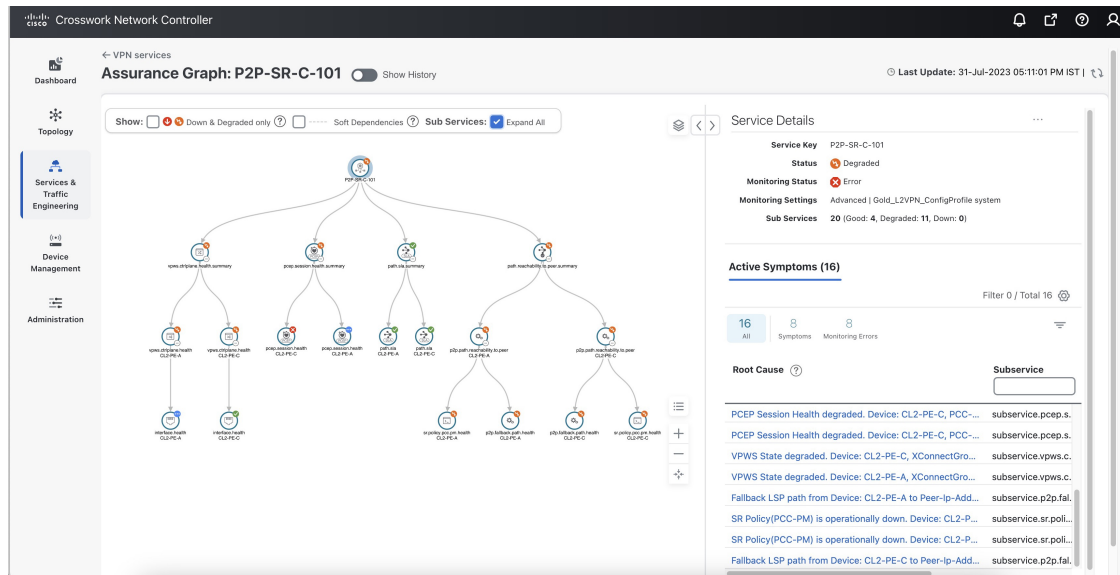
Note For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status will remain in the degraded state with a monitoring status as Monitoring error. Stop and restart the service monitoring to recover from this error.

To identify the root causes using Assurance Graph, do the following:

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

Step 2 In the **Actions** column, click [...] for the required degraded service and click **Assurance Graph**. The service assurance dependency graph view 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.

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

Step 3   By default, the Assurance Graph displays a concise view with only the service and the top level dependencies (feature 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 4   Select a degraded subservice in the Assurance Graph. The Subservice Details panel appears with subservice metrics, as well as 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.

Note   At the top left of the service assurance dependency graph, check the **Down & Degraded only** or **Soft Dependencies** check boxes to further isolate the subservices. 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.

Step 5 Inspect the **Active Symptoms** and **Impacted Services** information, and the root causes associated with the degraded service to determine the issues that may need to be addressed to maintain a healthy setup.

Related Information

- To view the Active Symptoms and Root Causes, see Identify Active Symptoms and Root Causes of a Degraded Service, on page 35.

- To identify the issues with the degraded services within a specific time range, use the Last 24Hr Metrics. For details, see Identify Root Causes Using Last 24Hr Metrics, on page 43.

- To identify a service health issue by examining the collection jobs, see View Collection Jobs, on page 47.

Identify Root Causes Using Last 24Hr Metrics

You can utilize the Last 24Hr Metrics to identify the issues with the degraded services within a specific time range. By isolating the issues within a specific time range, you can drill down on the details that may have caused the degraded (or down) service that can lead to troubleshooting the service or the node to address detailed symptoms.

Before you begin

- Ensure that service health monitoring is enabled for the service you want to analyze. For details, see Start Service Health Monitoring, on page 11.

Note For an L3VPN service being monitored, if a provider and devices are deleted, and then added again, the monitoring status will remain in the degraded state with a monitoring status as Monitoring error. Stop and restart the service monitoring to recover from this error.

Step 1 From the main menu, choose **Services & Traffic Engineering** > **VPN Services**. The service assurance dependency graph 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 degraded service and click **Assurance Graph**. The service assurance dependency graph 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.
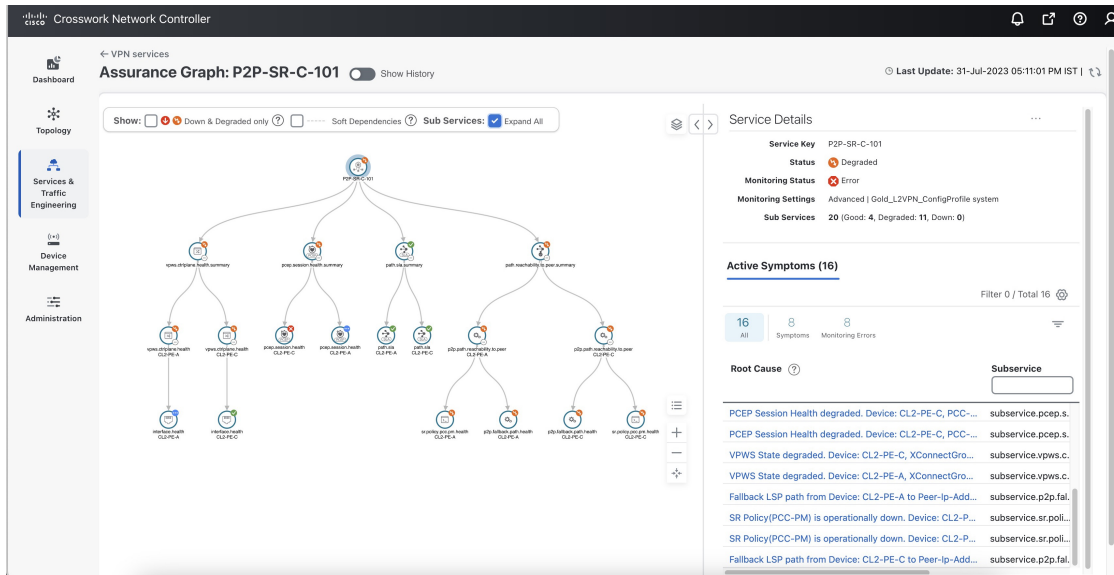
Step 3   At the top of the page, click the **Show History** mode toggle. The historical Date Range graph appears. This graph shows different ranges of historical health service monitoring details from one day (1d) up to sixty days (60d). When you hover over an event on the Date Range graph, a tool tip with information about that event appears (such as date and time of the event, and number of symptoms).
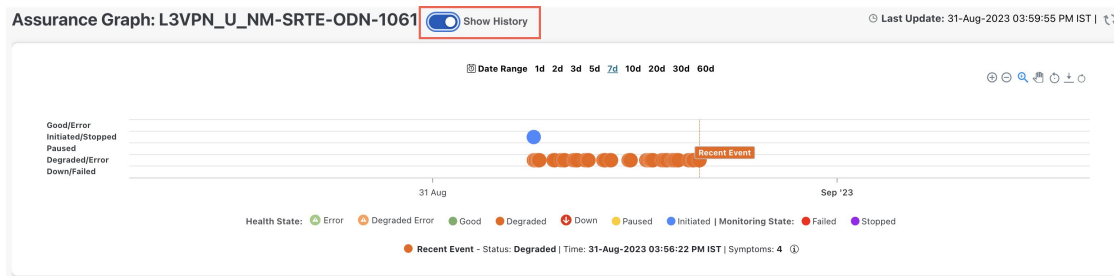
Step 4   Review the Root Cause information by clicking a particular event in the graph. The Service Details panel reloads, showing the active symptoms and the root causes associated with the event. Columns can be resized using your mouse or you can select the gear icon to deselect or select columns you want to appear.

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

Service Details

Service Key	L3VPN_U_NM-SRTE-ODN-1061
Status	🔶 Degraded
Monitoring Status	❌ Error
Monitoring Settings	Advanced | Gold_L3VPN_ConfigProfile custom
Sub Services	27 (Good: **19**, Degraded: **5**, Down: **0**)

Symptoms (4)

4 All	2 Symptoms	2 Monitoring Errors	Total 4 ⚙ ≡

Root Cause ⓘ   **Subservice**

Unable to get feed from device for metric(s): metric.inter...   subservice.interfa...

Unable to get feed from device for metric(s): metric.inter...   subservice.interfa...

eBGP Session to neighbor 10.10.10.238 is not up for D... 📊 Last 24Hr Metrics

eBGP Session to neighbor 10.10.10.238 is not up for Device: CL2-PE-A, Vrf:
L3VPN_U_NM-SRTE-ODN-1061

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) up to sixty days (60d).

Note   At the top-right of the Date Range graph, select the appropriate icons to either zoom in or out, horizontally scroll through the date ranges, or refresh the graph to go back 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. Events that are consecutive 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.

Step 6   Check the **Down & Degraded Only** check box at the top-left corner of the Assurance graph to show only the Subservices which 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 Assurance graph. 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 Assurance graph to zoom in or out on sub-services mapped. Select the **?** to view the Link Color Legend that explains all of the icons, symbols, badges, and colors and their definitions.

Step 7   Select the degraded subservice in the Assurance graph to show the subservice details.

Step 8 Click the **Symptoms** tab to show any root causes for the service health details that are displayed 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 and in the Actions column, click ⎕ for the degraded service in the list 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, different sessions states (such as active, idle, failed if applicable) 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.
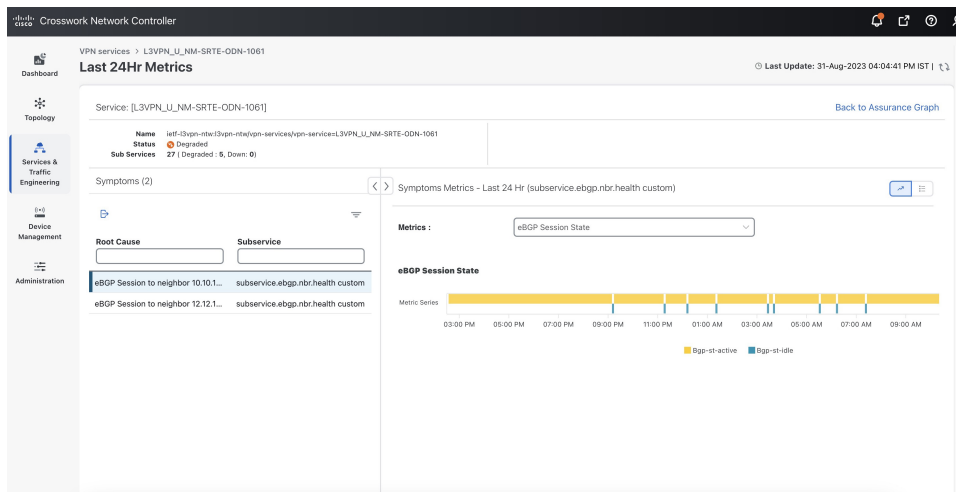
Related Information

- To view the Active Symptoms and Root Causes, see Identify Active Symptoms and Root Causes of a Degraded Service, on page 35.

- To monitor the VPN services using Assurance Graph capabilities and inspect any services or related nodes that are degraded, see Identify Root Causes Using Assurance Graph, on page 40.

- To identify a service health issue by examining the collection jobs, see View Collection Jobs, on page 47.

View the Devices Participating in the Service

When a device or interface related subservice degrades, the corresponding devices display an orange icon in the topology view. To view the devices participating in the services, do the following:

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

Step 2 Click a service that shows as degraded. The topology map is updated, isolating the corresponding devices participating in that service.

Step 3 At the top-left of the service assurance dependency graph view, select the **Show Participating Only** check box so that the topology map only shows the devices participating in the service.
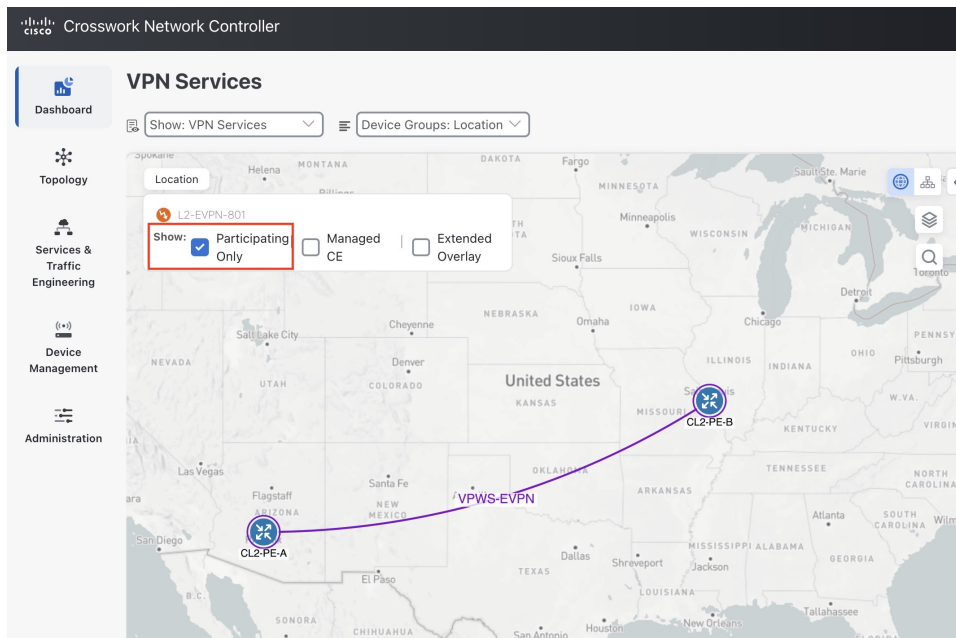
Step 4   Hover your mouse over the device icons and review the popup information relating to its Reachability State, Host Name, Node IP, and Type.

The devices that are healthy may show an orange badge to indicate that there are device or interface related subservices underneath that are not healthy. This ensures that unhealthy subservices are easily visible and can be identified from the topological view even if the device itself is healthy. After examining the Service Details for a device, for example, a condition, such as the CPU is low on a subservice node, helps to take the necessary steps to address the unhealthy subservice.

View Collection Jobs

The **Parameterized Jobs** tab on the **Collection Jobs** page displays all active jobs created by the Service Health application in Crosswork Network Controller.

Note   If Service Health is not deployed, this page will not contain any data.

Crosswork Network Controller enables you to view Parameterized Jobs, which are template-based collection jobs that support a large number of tasks, including CLI collection jobs. This feature is particularly useful for troubleshooting collection job issues, as it allows you to examine the details of individual devices. Devices are identified by their Context ID (protocol), indicating whether the jobs are gNMI, SNMP, or CLI-based.

Step 1   From the main menu, choose **Administration** > **Collection Jobs**.

The Collection Jobs page appears.

Step 2   Click the **Parameterized Jobs** tab.

Step 3   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 4   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 5   Click **Export**.

Step 6   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 7   Review the exported .csv file for collection job details and the possible cause of the degraded device.

Configure Additional Storage

This section explains the following topic:

-

Configure Additional External Storage

Crosswork Service Health provides internal storage of monitored data up to a maximum limit of 50 GB. The data includes the VPN service status at the time of storage and historical data about the service. This data is stored by Service Health on your system in the tar.gz archive file format. Each tar.gz file represents an EoS (Event of Significance). Service Health uses this data to display it visually in the Crosswork Network Controller UI when you click on an EoS.

When the storage reaches 70% capacity, Crosswork Network Controller generates an alarm prompting you to configure external storage. Crosswork Service Health automatically deletes the last recently used files when 80% of 50 GB storage capacity is reached.

By leveraging external storage, all existing internal storage data will be automatically moved to the external cloud storage and your internal storage will act as the cache storage.

You can use an Amazon Web Services (AWS) cloud account to configure external storage in the cloud. Only AWS S3, which is object storage, is supported.

After you configure AWS storage, only 80% of the 50 GB space or 100,000 files are stored locally in Crosswork Network Controller. The last recently used files are automatically moved to AWS.

Before you begin

You must have an AWS cloud account set up so to configure the external storage.

Step 1   From the main menu, choose **Administration** > **Settings** and click the **Storage Settings** tab.

Step 2 With the Overview tab selected, click **Configure** under the **External Storage** section. The Configuration page appears with the Data Storage Type and S3 Provider fields pre-populated with AWS.

Step 3 Provide your AWS authentication information for all of the required fields (such as Access Key, Secret Key, End Point, and so on).

Step 4 Check the **Copy Local Data** check box if you want all files, previously stored in the local cache, to be bulk copied to the external storage. This action will allow for incremental upload of the new files.

> **Note** This option is a one-time action when moving from only maintaining local storage and moving to external storage. This action also helps to improve the application performance.

> **Note** The Expiry Period refers to the number of days that historical data files will be stored before being deleted. For example, if the Expiry Period is set to 1, the files will be deleted two days later, at midnight of the operational time zone of the second day.

Step 5 Click **Test & Save**.

Step 6 To check the health of your storage setup, click the **Diagnostics** tab and click **Run Test**.

By running a test, you can review the external storage diagnostics such as bandwidth, latency, and multiple access test details to help identify the possible storage performance issues.

Customize Heuristic Packages

This section explains the following topics:

About Heuristic Packages

Crosswork Service Health uses Heuristic Packages as the core logic to monitor and report the health of services. Heuristic Packages define a list of rules, configuration profiles, supported subservices and associated metrics for every service type.

To access the Heuristic Packages, from the Main Menu, choose **Administration** > **Heuristic Packages**. The **Heuristic Packages** page has two tabs - **System** and **Custom**. The default set of Heuristic packages provided with Service Health are called system packages. These packages are available in the **System** tab. System packages cannot be modified. To customize a package to match your preferences you need to export, modify, and then import it back as a custom package. You can view the custom packages in the **Custom** tab.

Expand each section in this page to get more details on the services monitored and the thresholds used to generate alerts. You can also hover your mouse over the information **i** icon for finer details and definitions.
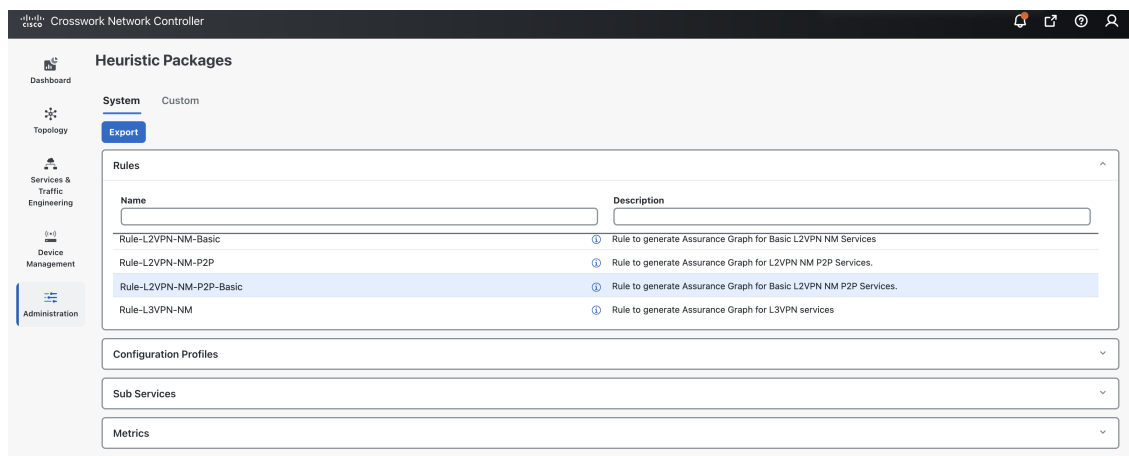
- **Rules**: Rules are used to structure services and the dependant sub-services and metrics within a specific service type. Dependencies within these rules help define the sub-services and the metrics that will

required for generating the data to assess the health of the service. A service can depend on an individual sub-service, a list of sub-services of the same type, or sub-services of different types.

For list of rules supported in Service Health, see Basic and Advanced Monitoring Rules, on page 59.

- **Configuration Profiles**: Configuration Profiles define threshold values that act as benchmarks for assessing the health of the service. By setting specific threshold values, Configuration Profiles establish the criteria for determining when a monitored parameter is within an acceptable range or deviates from the norm.

  Service Health with system heuristics package includes two configuration profiles - Silver and Gold for each of the service types (L2VPN and L3VPN). You can choose a profile option that aligns with your specific monitoring requirements. For instance, a Silver profile has more lenient thresholds compared to a Gold profile. You can create custom configuration profiles as needed.

- **Sub Services**: Sub services are characterized by a list of metrics to fetch and a list of computations to apply to these metrics in order to produce a health status and associated symptoms for the service.

  For example, the sub-service *subservice.evpn.health* monitors EVPN health. It is dependant on the metric *metric.l2vpn.xconnect.pw.state*. It evaluates an expression to check if *evpn_state* is **Up** and raises a symptom if degraded.

  For list of sub-services supported in Service Health, see Reference - Supported Subservices, on page 83.

- **Metrics**: Metrics define the operational data that should be fetched from different device types. Service Health uses a metric engine to map device-independent metrics to device-specific implementations, supporting multiple combinations of platforms and operating systems.

  For example, fetching the metric *resource.cpu* depends on the device type. For Cisco IOS XR devices, it uses Model-Driven Telemetry (MDT), while for Cisco IOS XE devices, it relies on CLI scraping using the command `show platform resources`.

In essence, there is a hierarchical relationship between Rules, Configuration Profiles, Sub services, and Metrics. Specifically, each Rule is mapped to a type of service, and depends on a number of sub-services to compute service health, sub-services use metrics and configuration profiles set threshold values for the metrics. Based on the values defined in these files, Service Health assesses the health of the service and builds the Assurance Graph.

Here is an Example that illustrates the hierarchical relationship between Rules, Sub services, and Metrics.

Customizing Heuristic Packages

The Heuristic Package (HP) bundled with Service Health functions as an assurance model for monitoring L2VPN and L3VPN services. However, the configurations of underlay and overlay networking services may vary across deployments. While the Heuristic Package can adapt automatically to certain configuration variations, other variations cannot be seamlessly absorbed. Examples of such variations include changes in the service function pack model, the introduction of a new device type in VPN service deployment, or the introduction of new network features requiring monitoring. In these scenarios, customization may be necessary for configuration profiles, rules, metrics, or sub-service class definitions.

Refer to the section Build a Custom Heuristic Package, on page 55 for a basic example of how you can create a custom package by customizing the configuration profile for a service. For further details and assistance in building a custom package based on rules, metrics, or sub-services, reach out to Cisco's Customer Experience (CX) team or your Cisco account team.

Build a Custom Heuristic Package

The procedure outlined below provides the steps for building a custom Heuristic Package by adjusting the threshold for acceptable CPU usage on the device (`CPU_THRESHOLD_MAX`) within the `Gold_L2VPN_ConfigProfile` of the Heuristic Package for L2VPN services.

Step 1 From the main menu, choose **Administration** > **Heuristic Packages**. The Heuristic Packages page opens with **System** and **Custom** tabs.

Step 2 Click the **System** tab and then click **Export**.

The `exportAPI.tar.gz` package gets downloaded to your system.

Step 3 Untar the `exportAPI.tar.gz` file, and you will get a `system` folder.

Step 4 In the `ConfigProfile` folder, open the `Gold_L2VPN_ConfigProfile-system.json` file.

Step 5 Make the following changes in the file:

a) Change the `namespace` attribute for the profile to `custom`.

b) Search for `CPU_THRESHOLD_MAX` and update the threshold value to 80.

```
{
  "name": "Gold_L2VPN_ConfigProfile",
  "namespace": "custom",
  "version": "1.0",
  "description":  "Thresholds to use for Gold L2VPN services",
  "rules": [
    {
      "name": "Rule-L2VPN-NM",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-NM-P2P",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-NM-Basic",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-NM-P2P-Basic",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-MP-Basic",
      "namespace": "system"
    },
    {
      "name": "Rule-L2VPN-MP",
      "namespace": "system"
    }
  ],
  "values": {
    "MAX_ACCEPTABLE_IN_OUT_PKT_DELTA": {
      "description": "Max allowed difference between packets received and packets transmitted",
      "type": "VAL_INT",
      "intVal": {
        "unit": "NA",
        "val":  100
      }
```

```
    },
    "VPN_INTF_PKT_ERROR_THRESHOLD": {
     "description": "Acceptable delta of in(or out) packet errors expected between polling intervals",

      "type": "VAL_INT",
      "intVal": {
        "unit": "NA",
        "val": 10
      }
    },
    "VPN_INTF_PKT_DISCARDS_THRESHOLD": {
      "description": "Acceptable delta of in(or out) packet discards expected between polling
intervals",
      "type": "VAL_INT",
      "intVal": {
        "unit": "NA",
        "val": 10
      }
    },
    "LATENCY_RT_THRESHOLD": {
      "description": "High Threshold for latency health checks",
      "type": "VAL_INT",
      "intVal": {
        "unit": "MSEC",
        "val": 500
      }
    },
    "JITTER_RT_THRESHOLD": {
      "description": "Threshold for acceptable jitter",
      "type": "VAL_FLOAT",
      "floatVal": {
        "unit": "MSEC",
        "val": 80
      }
    },
    "PACKET_LOSS_THRESHOLD": {
      "description": "Threshold for acceptable packet loss rate",
      "type": "VAL_FLOAT",
      "floatVal": {
        "unit": "PERCENT",
        "val": 1
      }
    },
    "SRPM_DELAY_THRESHOLD": {
      "description": "High Threshold for SR-PM latency health checks",
      "type": "VAL_INT",
      "intVal": {
        "unit": "MSEC",
        "val": 200

      }
    },
    "CPU_THRESHOLD_MAX": {
      "description": "Threshold for acceptable CPU usage on the device.",
      "type": "VAL_FLOAT",
      "floatVal": {
        "unit": "PERCENT",
        "val": 80
      }
    },
    "MEMFREE_THRESHOLD_MIN": {
      "description": "Threshold for minimum free memory to be available on the device.",
      "type": "VAL_FLOAT",
      "floatVal": {
```

```
        "unit": "BYTES",
        "val":  2000000000
      }
    }
  }
}
```

Step 6 Save the file once you have finished making the changes.

Step 7 Create a compressed tar.gz file from the `system` folder.

What to do next

Import the custom Heuristic Package in Crosswork Network Controller. See .

Import Custom Heuristic Packages

Follow this procedure to import the custom heuristic package in Crosswork Network Controller.

Step 1 From the main menu, choose **Administration** > **Heuristic Packages**. The Heuristic Packages page opens with **System** and **Custom** tabs.

Step 2 Click the **Custom** tab and then click **Import**. The **Import Heuristic Packages** dialog box appears.

Step 3 Click **Browse** to locate the custom package (*.tar.gz file) on your system.

Import Heuristic Package

⚠ System performance might be impacted during Heuristic package import due to high server resource consumption.

File Name

| | Browse |

| Preview ⌄ |

☐ I acknowledge that after importing heuristic packages, already monitored services cannot be edited/paused. Existing monitored services need to be restarted for new Heuristic package to take into effect.

Cancel   Import

Step 4 Select your custom package and click **Preview** to review the details of the package to be imported. Further information on the package's Rules, Configuration Profiles, Sub Services, and Metrics appears.

Note Your system performance might be impacted during heuristic package import due to high server resource consumption.

Select each option to preview the details of the custom package. Crosswork Network Controller will validate the package and display an error message if any issues exist. If there are no validation errors, Crosswork Network Controller will display a success message.

Step 5 Select the check box to acknowledge the warning and click **Import**. The package gets imported in Crosswork Network Controller and appears in the **Custom** tab in the **Configuration Profiles** section.

What to do next

To monitor services with custom heuristic packages, stop monitoring the service first. Start monitoring the service again by selecting the custom package and click **Start Monitoring**. See Step 4 in the procedure Start Service Health Monitoring, on page 11 for more information.

Reference - Basic Monitoring and Advanced Monitoring Rules

This section explains the following topics:

Basic and Advanced Monitoring Rules

Crosswork Service Health monitoring provides two options for monitoring: Basic Monitoring and Advanced Monitoring. The table below outlines the monitoring functions of each rule and sub-services, as well as the metric dependencies for both Basic and Advanced monitoring rules included in the system-defined Heuristic Package:

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-NM-Basic	• Checks the health of the VPWS xconnect state. • Monitors the health of the device: CPU and memory utilization.	subservice.device.health subservice.vpws.ctrlplane.health metric.l2vpn.xconnect.state metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-NM (Advanced)		

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices

	• Checks the health of the VPWS or EVPN xconnect state.
	
	• Monitors the health of the device: CPU and memory utilization.
	
	• Monitors the delta between received and transmitted packets between VPN interfaces and Pseudo-wire.
	
	• Monitors Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds.
	
	• Monitors the health status of RSVP tunnel. Subservice health will be marked as 'degraded' in either of the below scenarios:
	
	   • FRR is configured, but backup is not ready.
	
	   • FRR backup is active (primary failed and traffic is flowing over FRR backup).
	
	• Health check for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard.
	
	• Checks BGP Neighbor session health.
	
	• Checks whether all BGP EVPN next hops for a given L2VPN service are reachable over LSP.
	
	• Monitors PCEP session state to all the peers configured on this device.
	
	• Checks path reachability between two endpoints.
	
	• SR Policy (PCC initiated)
		subservice.bgp.nbr.health
		
		subservice.bgp.evpn.nexthop.health
		
		subservice.device.health
		
		subservice.evpn.health (one for each endpoint)
		
		subservice.fallback.path.health
		
		subservice.interface.health (one for each interface)
		
		subservice.l2vpn.y1731.health
		
		subservice.path.reachability.to.peer (local to remote and remote to local)
		
		subservice.path.sla
		
		subservice.pcep.session.health (one for each endpoint device)
		
		subservice.plain.lsp.path.health
		
		subservice.sr.policy.pce.health (one for each endpoint)
		
		subservice.vpws.ctrlplane.health (local, remote)
		
		subservice.path.reachability.to.peer
		
		subservice.fallback.path.health
		
		subservice.mpls.rsvpte.tunnel.pm.health
		
		subservice.l2vpn.y1731.health
		
		subservice.vpws.ctrlplane.health
		
		subservice.interface.health
		
		subservice.device.health
		
		subservice.interface.health.summary
		
		subservice.path.sla.summary
		
		metric.bgp.router.id
		
		metric.cef.route.labeled.lsp
		
		metric.l2vpn.xconnect.ac.state
		
		metric.l2vpn.xconnect.pw.state
		
		metric.l2vpn.xconnect.state
		
		metric.device.xconnect.ac.in.packets
		
		metric.device.xconnect.pw.out.packet
		
		metric.l2vpn.y1731.connect.cross.check. status
		
		metric.interface.oper

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling. • Checks whether LSP path exists (in default VRF) towards the given destination device.	metric.interface.in.errors metric.device.cpu.load metric.device.memory.free
Rule-L2VPN-NM-P2P-Basic	• Checks the health of the VPWS xconnect state. • Monitors the health of the device: CPU and memory utilization.	subservice.device.health subservice.vpws.ctrlplane.health
Rule-L2VPN-NM-P2P (Advanced)	• Checks the health of the VPWS xconnect state. • Monitors the health of the device: CPU and memory utilization. • Checks the health for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard. • Monitors Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds. • Monitors the LSP path to the peer VPN node. • Monitors path reachability between two endpoints. • Monitors LSP path (in default VRF) towards the given destination IP address. • Monitors PCEP session state to all the peers configured on this device. • Checks the SR Policy (PCC initiated) health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling.	subservice.device.health subservice.interface.health (one for each interface) subservice.l2vpn.y1731.health subservice.p2p.fallback.path.health subservice.p2p.path.reachability.to.peer (path reachability between endpoints) subservice.p2p.plain.lsp.path.health subservice.path.sla subservice.pcep.session.health (one for each endpoint device) subservice.sr.policy.pcc.health subservice.sr.policy.pce.health (one for each endpoint) subservice.vpws.ctrlplane.health (local, remote) metric.cef.route.labeled.lsp metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.l2vpn.xconnect.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-MP-Basic	• For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Monitors the health of the device • Monitors bridge domain state on a given endpoint.	subservice.device.summary subservice.bridge.domain.summary subservice.device.health subservice.bridge.domain.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-MP (Advanced)		

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	• For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Monitors the health of the device. • Groups together all the PCEP session health subservices. • Monitors PCEP session state to all the peers configured on this device. • Groups together all the device subservices. • Checks BGP Neighbor health. • Monitors whether any routes are present for the given Bridge Domain. • Groups together all the bridge domain subservices. • Monitors bridge domain state on a given endpoint. • Subservice to reflect interface health. • Groups together all the transport subservices. • SR Policy health status reflecting SR-PM SLA (if configured). Admin and Oper should be up. Oper should have stayed up since last polling. Delay and Variance should meet SLA if SR-PM is configured to measure delay. Liveness should be up if SR-PM is configured for Liveness. • Monitors the policies deployed by the ODN.	subservice.device.summary subservice.device.health subservice.pcep.session.health.summary subservice.pcep.session.health subservice.evpn.summary subservice.bgp.nbr.health subservice.mac.learning subservice.bridge.domain.summary subservice.bridge.domain.state subservice.interface.health subservice.transport.summary subservice.sr.policy.pcc.pm.health subservice.sr.policy.pce.pm.health subservice.mpls.rsvpte.tunnel.pm.health subservice.l2vpn.sr.odn.policy.dynamic metric.device.memory.free (supports XR only) metric.device.cpu.load (supports XR only) metric.sr.te.pcc.peer.state (supports XR only) metric.sr.te.pcc.peer.addrs (supports XR only) metric.bgp.session.state (supports XR only) metric.bgp.neighbors.ipaddr.list (supports XR only) metric.mac.learning.nexthops (supports XR only) metric.l2vpn.bridge.ac.state (supports XR only) metric.l2vpn.bridge.ac.list (supports XR only) metric.l2vpn.bridge.domain.state (supports XR only) metric.interface.oper (supports both XR and XE) metric.interface.in.errors (supports both XR and XE) metric.interface.out.errors (supports both XR and XE) metric.interface.in.discards (supports both XR and XE)

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	• SR Policy health status that include SR-PM. Admin and Oper should be up, and Oper should have stayed up since last polling. Delay and Variance should meet SLA if SR-PM is configured to measure delay. Liveness should be up if SR-PM is configured for Liveness. • Monitors MPLS RSVP TE Tunnel Health. Admin, Oper should both be up and if FRR is configured, then backup path should be ready to pickup traffic when primary fails. If failover already happened to backup then health will be shown as degraded as there is no more redundancy in play. Delay should be considered if SR-PM is enabled. If delay is enabled, then variance will be considered.	metric.interface.out.discards (supports both XR and XE) metric.sr.policy.pcc.admin.state (supports XR only) metric.sr.policy.pcc.oper.state (supports XR only) metric.sr.policy.pcc.oper.up.time (supports XR only) metric.sr.policy.pm.delay.measurement (supports XR only) metric.sr.pm.delay (supports XR only) metric.sr.pm.variance (supports XR only) metric.sr.policy.pm.liveness.detection (supports XR only) metric.sr.pm.liveness.state (supports XR only) metric.sr.policy.pce.admin.state (supports XR only) metric.sr.policy.pce.oper.state (supports XR only) metric.sr.policy.pce.oper.up.time (supports XR only) metric.sr.policy.pce.ietf.policy.name (supports XR only) metric.sr.policy.pm.delay.measurement (supports XR only) metric.sr.pm.delay (supports XR only) metric.sr.pm.variance (supports XR only) metric.sr.policy.pm.liveness.detection (supports XR only) metric.sr.pm.liveness.state (supports XR only) metric.mpls.rsvpte.tunnel.oper.state (supports XR only) metric.mpls.rsvpte.tunnel.admin.state (supports XR only) metric.mpls.rsvpte.tunnel.frr.configured (supports XR only) metric.mpls.rsvpte.tunnel.frr.status (supports XR only)

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.mpls.te.pm.delay.measurement (supports XR only)
		metric.mpls.rsvp.te.delay (supports XR only)
		metric.mpls.rsvp.te.variance (supports XR only)
		metric.l2vpn.odn.sr.policies.list (supports XR only)
		metric.bgp.router.id (supports both XR and XE)
Rule-L3VPN-NM-Basic	• Reports the overall route connectivity health between the current PE device and its connecting CE device. • Monitors the health of the device: CPU and memory utilization.	subservice.ce.pe.route.health subservice.device.health
Rule-L3VPN-NM-Node-Basic	• Monitors and presents the health summary for each PE device and the respective features underneath.	subservice.l3vpn.vpn.node.summary subservice.vrf.route.health subservice.device.health

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L3VPN-NM (Advanced)	• For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Subservice, together with child subservices in L3VPN Rule, reports the overall route health between current PE device and its connecting CE device. • eBGP Session health • Subservice to reflect interface health. • Monitors the health of the device. • L3VPN Aggregator Subservice that reflects path reachability from given device, for a given vrf, to peer VPN sites. • Monitors both static and dynamically initiated policy. • Checks whether plain LSP route exists within given VRF towards given vpn ip-addresses. • Monitors PCEP session state to all the peers configured on this device. • Checks BGP Neighbor health.	

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		subservice.ce.pe.route.health.summary
		subservice.ce.pe.route.health
		subservice.ebgp.nbr.health
		subservice.interface.health.summary
		subservice.interface.health
		subservice.device.summary
		subservice.device.health
		subservice.vrf.path.reachability.to.peer. summary
		subservice.vrf.path.reachability.to.peers
		subservice.transport.summary
		subservice.dynamic.l3vpn.sr.policy
		subservice.vrf.plain.lsp.reachability
		subservice.pcep.session.health.summary
		subservice.pcep.session.health
		subservice.bgp.nbr.health.summary
		subservice.bgp.nbr.health
		subservice.bgp.evpn.nexthop.health
		subservice.bgp.nbr.health
		subservice.ce.pe.route.health
		subservice.device.health
		subservice.ebgp.nbr.health
		subservice.evpn.health
		subservice.fallback.path.health
		subservice.interface.health
		subservice.l2vpn.y1731.health
		subservice.p2p.fallback.path.health
		subservice.p2p.path.reachability.to.peer
		subservice.p2p.plain.lsp.path.health
		subservice.path.reachability.to.peer
		subservice.path.sla
		subservice.pcep.session.health
		subservice.plain.lsp.path.health
		subservice.sr.policy.pcc.health

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		subservice.sr.policy.pce.health
		subservice.vpws.ctrlplane.health
		subservice.vrf.path.reachability.to.peers
		subservice.vrf.plain.lsp.reachability
		subservice.bridge.domain.summary
		subservice.l3vpn.sr.odn.policy.dynamic
		subservice.l2vpn.sr.odn.policy.dynamic
		subservice.mac.learning
		subservice.mpls.rsvpte.tunnel.pm.health
		subservice.vrf.path.reachability.to.peer. summary
		subservice.path.sla.summary
		subservice.pcep.session.health.summary
		subservice.transport.summary
		subservice.interface.health.summary
		subservice.vpws.ctrlplane.health.summary
		subservice.bridge.domain.state
		metric.route.vrf.connected (supports XR and XR IPv6)
		metric.route.vrf.local (supports XR and XR IPv6)
		metric.bgp.vrf.session.state (supports XR only)
		metric.interface.oper (supports both XR and XE)
		metric.interface.in.errors (supports both XR and XE)
		metric.interface.out.errors (supports both XR and XE)
		metric.interface.in.discards (supports both XR and XE)
		metric.interface.out.discards (supports both XR and XE)
		metric.device.memory.free (supports XR only)
		metric.device.cpu.load (supports XR only)
		metric.l3vpn.sr.policies.list (supports XR and XR IPv6)

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.cef.vrf.route.prefix (supports XR and XR IPv6)
		metric.sr.te.pcc.peer.state (supports XR only)
		metric.sr.te.pcc.peer.addrs (supports XR only)
		metric.bgp.session.state (supports XR only)
		metric.bgp.neighbors.ipaddr.list (supports XR only)
		metric.bgp.route.l2vpn.evpn.nexthops
		metric.bgp.router.id
		metric.cef.route.labeled.lsp
		metric.bgp.session.state
		metric.bgp.neighbors.ipaddr.list
		metric.route.vrf.connected
		metric.route.vrf.local
		metric.device.memory.free
		metric.device.cpu.load
		metric.bgp.vrf.session.state
		metric.l2vpn.xconnect.pw.state
		metric.cef.route.labeled.lsp
		metric.bgp.router.id
		metric.interface.oper
		metric.interface.in.errors
		metric.interface.out.errors
		metric.interface.in.discards
		metric.interface.out.discards
		metric.l2vpn.y1731.connect.cross.check. status
		metric.l2vpn.y1731.connect.peer.mep.status
		metric.l2vpn.y1731.latency.rt
		metric.l2vpn.y1731.jitter.rt
		metric.l2vpn.y1731.pktloss.1way.sd
		metric.l2vpn.y1731.pktloss.1way.ds
		metric.cef.route.labeled.lsp
		metric.cef.route.labeled.lsp
		metric.device.xconnect.ac.in.packets

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.device.xconnect.pw.out.packets
		metric.device.xconnect.pw.in.packets
		metric.device.xconnect.ac.out.packets
		metric.sr.te.pcc.ipv4.peer.state
		metric.sr.te.pcc.ipv4.peer.addrs
		metric.cef.route.labeled.lsp
		metric.bgp.router.id
		metric.sr.policy.pcc.oper.state
		metric.sr.policy.pcc.oper.up.time
		metric.sr.policy.pcc.admin.state
		metric.sr.policy.pm.delay.measurement
		metric.sr.pm.delay
		metric.sr.pm.variance
		metric.sr.policy.pm.liveness.detection
		metric.sr.pm.liveness.state
		metric.sr.policy.pce.oper.up.time
		metric.sr.policy.pce.oper.state
		metric.sr.policy.pce.admin.state
		metric.l2vpn.xconnect.state
		metric.l2vpn.xconnect.ac.state
		metric.l2vpn.xconnect.pw.state
		metric.cef.vrf.route.prefix
		metric.l3vpn.odn.sr.policies.dynamic.list
		metric.l2vpn.odn.sr.policies.list
		metric.bgp.router.id
		metric.mac.learning.nexthops
		metric.mpls.rsvpte.tunnel.oper.state
		metric.mpls.rsvpte.tunnel.admin.state
		metric.mpls.rsvpte.tunnel.frr.configured
		metric.mpls.rsvpte.tunnel.frr.status
		metric.mpls.te.pm.delay.measurement
		metric.mpls.rsvp.te.delay
		metric.l2vpn.bridge.ac.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.l2vpn.bridge.ac.list
		metric.l2vpn.bridge.domain.state
Rule-L3VPN-NM-Node (Advanced)	Monitors and presents the health summary for each PE device and the respective features underneath.	subservice.l3vpn.vpn.node.summary
		subservice.vrf.route.health
		subservice.device.health
		subservice.interface.health
		subservice.pcep.session.health
		subservice.bgp.nbr.health

Example

The given example explains the relationship between the 'Rule-L2VPN-NM-P2P-Basic' and its dependent sub-services, specifically 'subservice.vpws.ctrlplane.health' and 'subservice.device.health'. Additonally, the sub-service definitions are also listed below to highlight the metric dependencies and symptoms generated by these sub-services.

Rule-L2VPN-NM-P2P-Basic

```
{
  "name": "Rule-L2VPN-NM-P2P-Basic",
  "namespace": "system",
  "id": "Rule-L2VPN-NM-P2P-Basic system",
  "description": "Rule to generate Assurance Graph for Basic L2VPN NM P2P Services.",
  "matchCriteria": [
    {
      "configSource": "SOURCE_TYPE_NSO",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
"//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-svc-type[text()='vpn-common:t-ldp']",

      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NSO",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression": "//flat-L2vpn/service-type[text()='p2p']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NSO",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
```

```
                    "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='vpn-common:t-ldp']",

              "matchPrefix": "",
              "matchParams": []
            },
            {
              "configSource": "SOURCE_TYPE_NSO",
              "configSubSource": [
                "SUBSOURCE_SERVICE_CONFIG"
              ],
              "matchType": "MATCH_TYPE_XPATH",
              "matchExpression": "//vpn-service[not(//bridge-group)]/vpn-type[contains(text(),
      ':mpls-evpn')]",
              "matchPrefix": "",
              "matchParams": []
            },
            {
              "configSource": "SOURCE_TYPE_NSO",
              "configSubSource": [
                "SUBSOURCE_SERVICE_CONFIG"
              ],
              "matchType": "MATCH_TYPE_XPATH",
              "matchExpression":
      "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='x:vpws']",

              "matchPrefix": "",
              "matchParams": []
            },
            {
              "configSource": "SOURCE_TYPE_NSO",
              "configSubSource": [
                "SUBSOURCE_SERVICE_CONFIG"
              ],
              "matchType": "MATCH_TYPE_XPATH",
              "matchExpression":
      "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='ietf-vpn-common:vpws']",

              "matchPrefix": "",
              "matchParams": []
            }
          ],
          "dependencies": [
            {
              "name": "VPWS-ControlPlane-Health-Summary",
              "id": "subservice.vpws.ctrlplane.health.summary system",
              "ssClass": "subservice.vpws.ctrlplane.health.summary",
              "namespace": "system",
              "type": "DEP_TYPE_NON_LIST",
              "optional": false,
              "paramExtractionMechanism": {
                "mode": "EXTRACT_MODE_XPATH",
                "name": "",
                "namespace": "",
                "version": "",
                "validationHash": "0",
                "pluginMethod": "",
                "extractedParams": [],
                "nativeMethod": ""
              },
              "parameters": [
                {
                  "name": "vpnServiceId",
                  "iterator": false,
                  "defaultValue": "",
```

```
                    "extractionMethod": "DEP_PARAM_XPATH",
                    "extractionDetails": [
                      {
                        "description": "",
                        "extractValue": "//vpn-service/vpn-id"
                      },
                      {
                        "description": "Flat Model",
                        "extractValue": "//flat-L2vpn[/flat-L2vpn-p2p]/key"
                      }
                    ]
                  }
                ],
                "subDependencies": [
                  "VPWS-ControlPlane-Health-Local-Site",
                  "VPWS-ControlPlane-Health-Remote-Site"
                ],
                "softSubDependencies": []
              },
              {
                "name": "VPWS-ControlPlane-Health-Local-Site",
                "id": "subservice.vpws.ctrlplane.health system",
                "ssClass": "subservice.vpws.ctrlplane.health",
                "namespace": "system",
                "type": "DEP_TYPE_NON_LIST",
                "optional": false,
                "paramExtractionMechanism": {
                  "mode": "EXTRACT_MODE_XPATH",
                  "name": "",
                  "namespace": "",
                  "version": "",
                  "validationHash": "0",
                  "pluginMethod": "",
                  "extractedParams": [],
                  "nativeMethod": ""
                },
                "parameters": [
                  {
                    "name": "device",
                    "iterator": false,
                    "defaultValue": "",
                    "extractionMethod": "DEP_PARAM_XPATH",
                    "extractionDetails": [
                      {
                        "description": "",
                        "extractValue": "//vpn-nodes/vpn-node[1]/vpn-node-id"
                      }
                    ]
                  },
                  {
                    "name": "groupName",
                    "iterator": false,
                    "defaultValue": "",
                    "extractionMethod": "DEP_PARAM_XPATH",
                    "extractionDetails": [
                      {
                        "description": "",
                        "extractValue": "//vpn-service/vpn-id"
                      },
                      {
                        "description": "Flat Model",
                        "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/local-site/xconnect-group-name"
                      }
                    ]
```

```
          },
          {
            "name": "xconnectName",
            "iterator": false,
            "defaultValue": "",
            "extractionMethod": "DEP_PARAM_XPATH",
            "extractionDetails": [
              {
                "description": "",
                "extractValue": "//vpn-service/vpn-id"
              },
              {
                "description": "Flat Model",
                "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/local-site/xconnect-group-name"
              }
            ]
          }
        ],
        "subDependencies": [],
        "softSubDependencies": [
          "device1"
        ]
      },
      {
        "name": "VPWS-ControlPlane-Health-Remote-Site",
        "id": "subservice.vpws.ctrlplane.health system",
        "ssClass": "subservice.vpws.ctrlplane.health",
        "namespace": "system",
        "type": "DEP_TYPE_NON_LIST",
        "optional": false,
        "paramExtractionMechanism": {
          "mode": "EXTRACT_MODE_XPATH",
          "name": "",
          "namespace": "",
          "version": "",
          "validationHash": "0",
          "pluginMethod": "",
          "extractedParams": [],
          "nativeMethod": ""
        },
        "parameters": [
          {
            "name": "device",
            "iterator": false,
            "defaultValue": "",
            "extractionMethod": "DEP_PARAM_XPATH",
            "extractionDetails": [
              {
                "description": "",
                "extractValue": "//vpn-nodes/vpn-node[2]/vpn-node-id"
              }
            ]
          },
          {
            "name": "groupName",
            "iterator": false,
            "defaultValue": "",
            "extractionMethod": "DEP_PARAM_XPATH",
            "extractionDetails": [
              {
                "description": "",
                "extractValue": "//vpn-service/vpn-id"
              },
              {
```

```
            "description": "Flat Model",
            "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/remote-site/xconnect-group-name"

          }
        ]
      },
      {
        "name": "xconnectName",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",
        "extractionDetails": [
          {
            "description": "",
            "extractValue": "//vpn-service/vpn-id"
          },
          {
            "description": "Flat Model",
            "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/remote-site/xconnect-group-name"

          }
        ]
      }
    ],
    "subDependencies": [],
    "softSubDependencies": [
      "device2"
    ]
  },
  {
    "name": "device1",
    "id": "subservice.device.health system",
    "ssClass": "subservice.device.health",
    "namespace": "system",
    "type": "DEP_TYPE_NON_LIST",
    "optional": false,
    "paramExtractionMechanism": {
      "mode": "EXTRACT_MODE_XPATH",
      "name": "",
      "namespace": "",
      "version": "",
      "validationHash": "0",
      "pluginMethod": "",
      "extractedParams": [],
      "nativeMethod": ""
    },
    "parameters": [
      {
        "name": "device",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",
        "extractionDetails": [
          {
            "description": "",
            "extractValue": "//vpn-nodes/vpn-node[1]/vpn-node-id"
          }
        ]
      }
    ],
    "subDependencies": [],
    "softSubDependencies": []
  },
  {
```

```
          "name": "device2",
          "id": "subservice.device.health system",
          "ssClass": "subservice.device.health",
          "namespace": "system",
          "type": "DEP_TYPE_NON_LIST",
          "optional": false,
          "paramExtractionMechanism": {
            "mode": "EXTRACT_MODE_XPATH",
            "name": "",
            "namespace": "",
            "version": "",
            "validationHash": "0",
            "pluginMethod": "",
            "extractedParams": [],
            "nativeMethod": ""
          },
          "parameters": [
            {
              "name": "device",
              "iterator": false,
              "defaultValue": "",
              "extractionMethod": "DEP_PARAM_XPATH",
              "extractionDetails": [
                {
                  "description": "",
                  "extractValue": "//vpn-nodes/vpn-node[2]/vpn-node-id"
                }
              ]
            }
          ],
          "subDependencies": [],
          "softSubDependencies": []
      }
    ],
    "softRootDependencies": [],
    "createTimestamp": "1697841637567500247",
    "updateTimestamp": "0",
    "monitoringType": "BASIC",
    "version": "1.1"
}
```

Sub service: 'subservice.vpws.ctrlplane.health'

```
{
    "id": "subservice.vpws.ctrlplane.health.summary system",
    "name": "subservice.vpws.ctrlplane.health.summary",
    "namespace": "system",
    "description": "Groups together all the VPWS Ctrlplane health subservices.",
    "params": [
      {
        "name": "vpnServiceId",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
      }
    ],
    "liveMetrics": {},
    "rootExpressions": [],
    "dynamicConfig": null,
    "symptom": null,
    "dependencies": [],
    "exprCid": "",
    "createTimestamp": "1697841637373426164",
    "updateTimestamp": "0",
    "tags": [],
    "version": "1.0"
```

```
        }

        {
          "id": "subservice.vpws.ctrlplane.health system",
          "name": "subservice.vpws.ctrlplane.health",
          "namespace": "system",
          "description": "check the health of the VPWS state",
          "params": [
            {
              "name": "device",
              "description": "",
              "type": "PARAM_TYPE_NON_LIST"
            },
            {
              "name": "groupName",
              "description": "",
              "type": "PARAM_TYPE_NON_LIST"
            },
            {
              "name": "xconnectName",
              "description": "",
              "type": "PARAM_TYPE_NON_LIST"
            }
          ],
          "liveMetrics": {},
          "rootExpressions": [
            {
              "evalExpression": "xconnect_state == 'up' && ac_state == 'up' && evpn_state == 'up'",

              "activateCondition": ""
            }
          ],
          "dynamicConfig": null,
          "symptom": {
            "formatString": "VPWS State degraded. Device: {device}, XConnectGroup: {groupName},
XconnectName: {xconnectName}",
            "level": "DEGRADED",
            "priority": 15,
            "condition": false
          },
          "dependencies": [
            {
              "type": "DEP_TYPE_METRIC",
              "label": "xconnect_state",
              "evalExpression": "metric.l2vpn.xconnect.state",
              "namespace": "",
              "symptom": null,
              "paramMap": {
                "device": "device",
                "groupName": "groupName",
                "xconnectName": "xconnectName"
              },
              "id": ""
            },
            {
              "type": "DEP_TYPE_METRIC",
              "label": "ac_state",
              "evalExpression": "metric.l2vpn.xconnect.ac.state",
              "namespace": "",
              "symptom": null,
              "paramMap": {
                "device": "device",
                "groupName": "groupName",
                "xconnectName": "xconnectName"
```

```
      },
      "id": ""
    },
    {
      "type": "DEP_TYPE_METRIC",
      "label": "evpn_state",
      "evalExpression": "metric.l2vpn.xconnect.pw.state",
      "namespace": "",
      "symptom": null,
      "paramMap": {
        "device": "device",
        "groupName": "groupName",
        "xconnectName": "xconnectName"
      },
      "id": ""
    }
  ],
  "exprCid": "",
  "createTimestamp": "1697841637370064741",
  "updateTimestamp": "0",
  "tags": [],
  "version": "1.0"
}
```

Sub service: 'subservice.device.health'

```
{
  "id": "subservice.device.health system",
  "name": "subservice.device.health",
  "namespace": "system",
  "description": "Monitor the health of the device.",
  "params": [
    {
      "name": "device",
      "description": "",
      "type": "PARAM_TYPE_NON_LIST"
    }
  ],
  "liveMetrics": {},
  "rootExpressions": [
    {
      "evalExpression": "cpu_healthy && memory_healthy",
      "activateCondition": ""
    }
  ],
  "dynamicConfig": null,
  "symptom": {
    "formatString": "Heavier than expected resource consumption on the Device: {device}",
    "level": "DEGRADED",
    "priority": 100,
    "condition": false
  },
  "dependencies": [
    {
      "type": "DEP_TYPE_EXPRESSION",
      "label": "cpu_healthy",
      "evalExpression": "ListElemsAverage(cpu_load) <= CPU_THRESHOLD_MAX",
      "namespace": "",
      "symptom": null,
      "paramMap": {},
      "id": ""
    },
    {
      "type": "DEP_TYPE_EXPRESSION",
      "label": "memory_healthy",
```

```
        "evalExpression": "ListElemsSum(memory_free) > MEMFREE_THRESHOLD_MIN",
        "namespace": "",
        "symptom": null,
        "paramMap": {},
        "id": ""
      },
      {
        "type": "DEP_TYPE_METRIC",
        "label": "cpu_load",
        "evalExpression": "metric.device.cpu.load",
        "namespace": "",
        "symptom": null,
        "paramMap": {
          "device": "device"
        },
        "id": ""
      },
      {
        "type": "DEP_TYPE_METRIC",
        "label": "memory_free",
        "evalExpression": "metric.device.memory.free",
        "namespace": "",
        "symptom": null,
        "paramMap": {
          "device": "device"
        },
        "id": ""
      }
    ],
    "exprCid": "",
    "createTimestamp": "1697841637256704609",
    "updateTimestamp": "0",
    "tags": [
      "DEVICE_SUBSERVICES"
    ],
    "version": "1.1"
}

{
    "id": "subservice.device.summary system",
    "name": "subservice.device.summary",
    "namespace": "system",
    "description": "Groups together all the Device subservices",
    "params": [
      {
        "name": "vpnServiceId",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
      }
    ],
    "liveMetrics": {},
    "rootExpressions": [],
    "dynamicConfig": null,
    "symptom": null,
    "dependencies": [],
    "exprCid": "",
    "createTimestamp": "1697841637260108075",
    "updateTimestamp": "0",
    "tags": [],
    "version": "1.0"
}
```

APPENDIX B

Reference - Supported Subservices

The following tables provide details of supported Service Health L2VPN/L3VPN flavors and associated subservices (for IOS XE and IOS XR devices).

Table 6: Supported VPN Services with Associated Subservices (for IOS XE Devices)

Supported VPN Services	Associated Subservices	Details
L2VPN Point to Point with SR underlay	• Path Reachability • Y.1731 Health • VPN Interface Health • Device Health • Summary (aggregator) nodes	IOS XE does not support SNMP/gNMI this subservice (CEF route; PCEP Sessi State; XConnect).
L2VPN Point to Point over MPLS LDP	• Path Reachability • Y.1731 Health • VPWS Control Plane health • VPN Interface Health • Device Health • Summary (aggregator) nodes	IOS XE does not support SNMP/gNMI this subservice (CEF route; XConnect).
L2VPN P2P Plain	• Path Reachability • Y.1731 Health • VPN Interface Health • Device Health • Summary (aggregator) nodes	IOS XE does not support SNMP/gNMI this subservice (CEF route; XConnect). **Note**: The reference to 'Plain' implies th traffic takes the IGP path and does not u like SR.

L3VPN SR	• Path Reachability	IOS XE does not support SNMP/gNMI col
	• CE-PE Route Health	this subservice (CEF route; PCEP Session S
	• eBGP Neighbor Health	is also not supported.
	• VPN Interface Health	
	• BGP Neighbor Health (DynExp)	
	• Summary (aggregator) nodes	

Table 7: Supported VPN Services with Associated Subservices (for IOS XR Devices)

Supported VPN Services	Associated Subservices
L2VPN EVPN SR	• Path Reachability
	• Fallback Enabled/Disabled (DynExp)
	• SR Policy – PCC
	• Path SLA
	• Y.1731 Health
	• VPWS Control Plane Health
	• VPN Interface Health
	• Device Health
	• EVPN Health
	• BGP Neighbor Health (DynExp)
	• BGP Nexthop Health (DynExp)
	• PCEP Session Health (DynExp)
	• SR Policy – PCE
	• Summary (aggregator) nodes

L2VPN EVPN Plain	• Path Reachability
	• Path SLA
	• Plain LSP Path Health (DynExp)
	• VPWS Control Plane health
	• VPN Interface Health
	• Device Health
	• EVPN Health
	• BGP Neighbor Health (DynExp)
	• BGP Nexthop Health (DynExp)
	• Summary (aggregator) nodes
	Note: The reference to 'Plain' implies that L2VPN/L3VPN traffic takes the IGP path and does not use any transports, like SR.
L2VPN Point to Point over RSVP	• Path Reachability
	• Fallback Enabled/Disabled
	• RSVP-TE Health
	• Path SLA
	• Y.1731 Health
	• VPWS Control Plane Health/Xconnect Health
	• VPN Interface Health
	• Device Health
L2VPN Point to Point with SR underlay	• Path Reachability
	• Fallback Enabled/Disabled
	• SR Policy – PCC
	• Path SLA
	• Y.1731 Health
	• VPWS Control Plane Health
	• VPN Interface Health
	• Device Health
	• PCEP Session Health (DynExp)
	• SR Policy – PCE
	• Summary (aggregator) nodes

L2VPN Point to Point over MPLS LDP	• Path Reachability
	• Fallback Enabled/Disabled
	• Path SLA
	• Y.1731 Health
	• VPWS Control Plane Health
	• VPN Interface Health
	• Device Health
	• Summary (aggregator) nodes
L2VPN P2P Plain	• Path Reachability
	• Plain LSP Path Health
	• Path SLA
	• Y.1731 Health
	• VPWS Control Plane Health
	• VPN Interface Health
	• Device Health
	• Summary (aggregator) nodes
	Note: The reference to 'Plain' implies that L2VPN/L3VPN traffic takes the IGP path and does not use any transports, like SR.
L3VPN SR	• CE-PE Route Health
	• eBGP Neighbor Health
	• VPN Interface Health
	• Device Health
	• Path Reachability
	• Vrf Plain LSP Path Health
	• PCEP Session Health (DynExp)
	• BGP Neighbor Health (DynExp)
	• Summary (aggregator) nodes
	• SR and SRv6 polices