Managing MPLS Transport Profile Services

This chapter describes the tasks required to get started using Prime Provisioning, Multiprotocol Label Switching (MPLS) Transport Profile (TP) services.

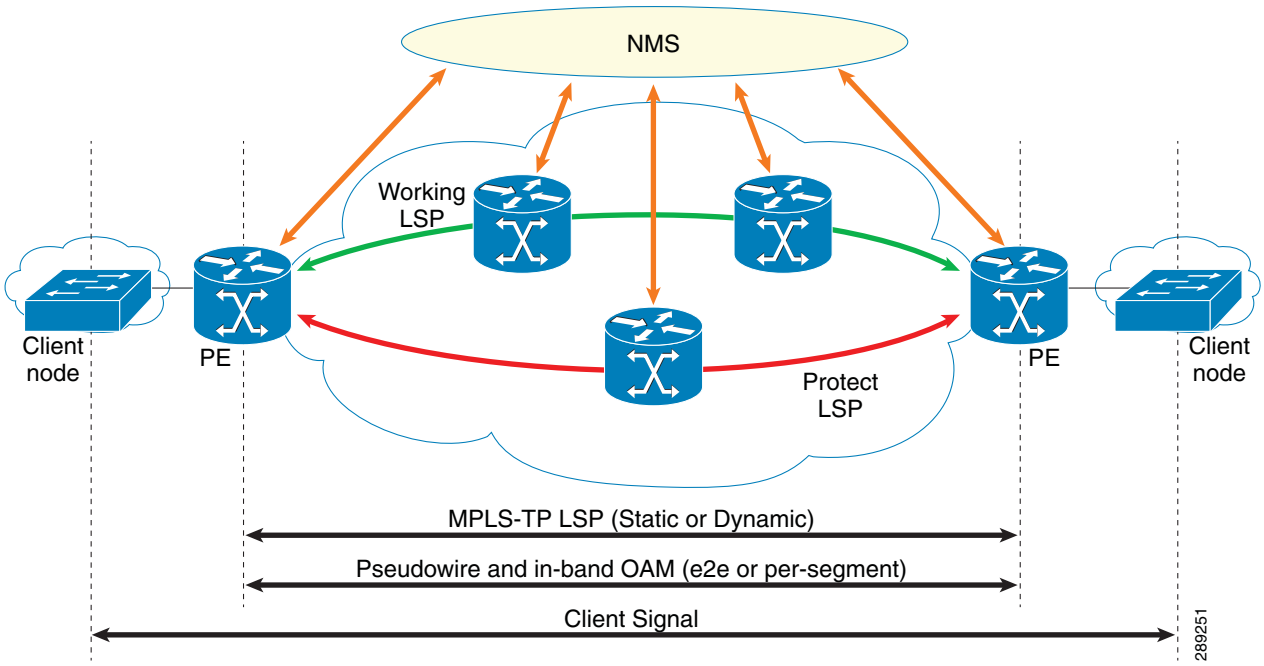
This section covers the following topics:

Introduction

MPLS-TP is a transport service (managed by Prime Provisioning) for a dynamic MPLS core.

In the current implementation of MPLS-TP, an MPLS-TP tunnel can be provisioned between two arbitrary nodes in an MPLS-TP enabled network. The provisioned tunnel can have one or two paths, a working and an optional protect label-switched path (LSP). The normal use case is for Prime Provisioning to automatically calculate the working and protect paths using a path selection algorithm that chooses MPLS-TP enabled links based on shortest path, and to provision the tunnel on the endpoints and all nodes traversed by the tunnel.

Figure 7-1 An MPLS-TP Enabled Network



Prerequisites and Limitations

The current release of Prime Provisioning involves certain prerequisites and limitations, which are described in the *Cisco Prime Provisioning 7.2 Installation Guide*, including general system recommendations.

Note that Internet Explorer 8 (IE8) will not show the calculated path graphically (as described in Creating an MPLS-TP Service Request, page 7-9) as IE8 offers no support for SVG display. Until IE9 is supported, a textual summary of the path can be used to review the path in IE8. IE9 (and other Prime Provisioning supported browsers) shows the calculated path graphically.

Changes performed to an operational device sometimes take time to reflect on Prime Network.

Polling is performed by Prime Network every 15 minutes (at least). In the duration of 1 to 15 minutes, polling is performed many times. Each poll collects different data (tunnels, labels, links, etc). Since all the information is not collected in a single poll, the time taken to reflect tunnel update, label update, links update varies in Prime Network.

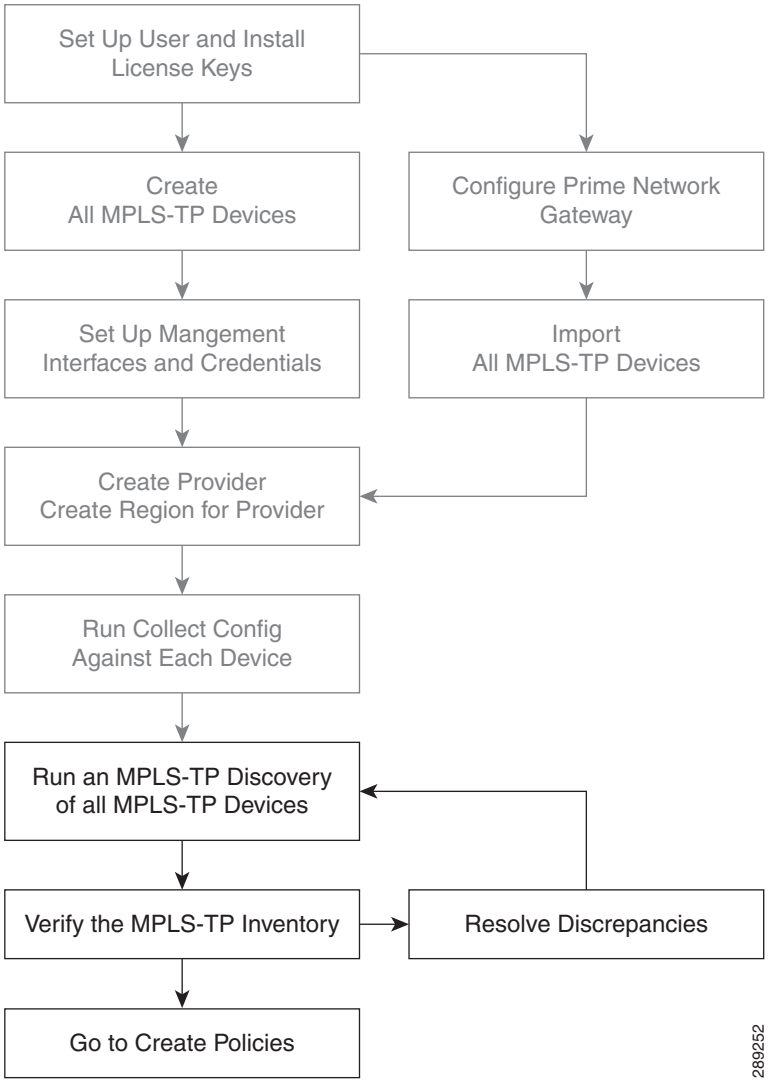
For supported device and OS information, refer to *Cisco Prime Provisioning 7.2 Supported Devices*.

Preconfiguration Process

The preconfiguration process sets up key parameters that enable the system to collect MPLS-TP network information and subsequently deploy MPLS-TP configurations on the chosen network.

The different steps in the preconfiguration process are provided in Figure 7-2.

Figure 7-2 Preconfiguration Process



Before commencing the preconfiguration process, MPLS-TP needs to be enabled on the network devices by making sure that the IP addresses used as devices' MPLS-TP IDs are accessible from the management station (this step is not supported by MPLS-TP). This is described in Other MPLS-TP Preconfiguration Requirements, page 7-4.

Setting up new user and installing license keys is described in *Cisco Prime Provisioning Administrator's Guide 7.2* and the other steps are covered in Setting Up Devices and Device Groups, page 2-1.

As a result, the Prime Provisioning user will need to wait some before running MPLS-TP Discovery after a device change in Prime Network.

Note When Prime Provisioning is integrated with Prime Network, it is required to import Prime Network certificate into Prime Provisioning Trust Store, which is described in Device Import Prerequisite, page 13-12.

See below for a description of specific MPLS-TP user roles.

The MPLS-TP-specific steps are as follows:

1. **Run an MPLS-TP Discovery Task**—Use Task Manager to discover the MPLS-TP network for a particular MPLS-TP provider to populate the repository with a view to creating primary and backup tunnels. (See Running MPLS-TP Discovery, page 7-5.)
2. **Verify the MPLS-TP Inventory**—Verify that the MPLS-TP Discovery task was successfully completed. This can be done in a couple of ways. (See Verifying the MPLS-TP Discovery Results, page 7-7.)

MPLS-TP Setup and Installation

Before setting up Prime Provisioning, the Prime Provisioning software must be installed. To do so, see the *Cisco Prime Provisioning 7.2 Installation Guide*.

To set up a new Prime Provisioning user, one or more users with a MPLS-TP role must be created. MPLS-TP roles are described in MPLS-TP User Roles, page 7-4 and for step by step instructions for creating user roles, refer to *Cisco Prime Provisioning 7.2 Administration Guide*.

For more information about Prime Provisioning licenses and for the procedure required to install licenses, refer to *Cisco Prime Provisioning 7.2 Administration Guide*.

MPLS-TP User Roles

Prime Provisioning currently supports two MPLS-TP roles, the MPLS-TPRole and MPLS-TPServiceOpRole. These two user roles behave similarly to the other roles in Prime Provisioning, for example the MPLSRole and the MPLSServiceOpRole found in MPLS.

They have the following permissions:

- MPLS-TPRole—full permission to manage the inventory (create, read, update, delete, and deploy MPLS-TP policies and service requests)
- MPLS-TPServiceOpRole—permission to deploy MPLS-TP service requests

For an explanation of how to work with roles, refer to *Cisco Prime Provisioning 7.2 Administration Guide*.

Other MPLS-TP Preconfiguration Requirements

Prior to performing MPLS-TP provisioning, perform the following additional configuration steps:

Step 1 Enable MPLS-TP on the device:

- Choose a global ID common to all devices (AS number, for example)
- Allocate a Router ID to each device.
- Configure MPLS-TP-related timers.

Step 2 Configure a range of statically defined MPLS labels to be used by MPLS-TP tunnels and static pseudowires.

Step 3 Enable MPLS-TP links to select which interfaces will form the links in the MPLS-TP topology:

- Give each interface an ID.
- Optionally configure a bandwidth pool on each interface.

Step 4 Create a BFD class to be used to monitor your MPLS-TP tunnels.

Running MPLS-TP Discovery

Prime Provisioning supports MPLS-TP discovery from IOS & IOS-XR devices when deployed together with Prime Network(s) (or) in a Prime suite.

As a prerequisite for running MPLS-TP discovery, all devices must be present and a Collect Config task must be run (see Collect Config from Files, page 12-4).

Prime Provisioning should be 'paired' with the Prime Network(s) by setting the gateway details of the Prime Network(s) in the Prime Provisioning DCPL properties **Inventory Import**. Multiple Gateways can be configured by separating the values with a comma. The order in which the Prime Networks are configured has an impact on the MPLS-TP discovery process. The instance mentioned first in the DCPL has the highest priority whereas the last has the least priority. For further details on setting the DCPL properties, refer to *Cisco Prime Provisioning 7.2 Administration Guide.* or see Appendix G, "Property Settings"

Note MPLS-TP discovery will update only the functional MPLS-TP links in the MPLS-TP routing diagram (Service Request Editor, Review Routing accordion).

The MPLS-TP discovery process discovers the following from the live network:

- TP enabled links
- MPLS Static label pools
- MPLS Static label pool usage
- BFD templates
- TP Router ID
- TP Global ID

When MPLS-TP discovery process runs, Prime Provisioning checks if the chosen device(s) are present in the Prime Network instances. If the device chosen for MPLS-TP discovery is present in multiple Prime Network instances, MPLS-TP discovery is performed based on the Prime Network instance priority. MPLS-TP information is collected from the highest priority, whereas the TP enabled links alone are collected from all the Prime Network instances.

Caution If an MPLS-TP network spans devices that are in multiple gateways, the link will not be discovered between devices that are in different gateways. To ensure a full network discovery, add border devices to both the Prime Network gateways, so that all TP links are visible at least in one Prime Network instance.

If the device chosen for MPLS-TP discovery is not present in the highest priority Prime Network, Prime Provisioning checks the next priority Prime Network instance 'paired' in the DCPL to discover the MPLS-TP information.

If the device chosen for MPLS-TP discovery is not present in any of the Prime Network instance, an error message is logged, "MPLS-TP Discovery for device "DeviceName" failed. Device not found in any of the gateways."

If the device chosen for MPLS-TP discovery has any directly connected neighbors and available in the Prime Provisioning inventory, MPLS-TP discovery is performed for the neighbors from the same Prime Network instance.

Prime Provisioning, in standalone mode (without Prime Network integration) supports CDP-based MPLS-TP discovery from IOS devices but this is deprecated.

MPLS-TP enabled devices should be added or created on Prime Provisioning Inventory by:

- Directly creating the devices on Prime Provisioning (or)
- Using the “Import” functionality available in the Prime Provisioning device creation page - where the device can be imported from Prime Network.

The MPLS-TP network is discovered using the **MPLS-TP Discovery** task. This populates the repository with the network topology in an automated way. Where possible, the discovery process will try to keep the repository consistent with the network, for example delete links which have been removed. In cases where this is not possible, for example if a link is in use, a log message will be recorded.

The necessary steps are described in the below sections:

- Creating an MPLS-TP Discovery Task, page 7-6
- Verifying the MPLS-TP Discovery Results, page 7-7

Creating an MPLS-TP Discovery Task

To create a MPLS-TP Discovery task on the MPLS-TP network, use the following steps:

Step 1 Choose **Operate > Task Manager**.

The Task Manager window appears.

Step 2 Create a new task by selecting **Create > MPLS-TP Discovery**.

The Create Task window appears.

Step 3 Make any desired changes to the auto-generated name and description text and click Next.

The **MPLS-TP Discovery** window appears.

Step 4 Select the devices through which the MPLS-TP network should be discovered.

Step 5 Click **Submit**.

The discovery process begins.

Step 6 Once the MPLS-TP discovery task is complete, the outcome will be documented in a log under:

Operate > Task Logs.

To run the MPLS-TP Discovery task immediately after the device creation navigate to:

Inventory > Devices > Create > Cisco Device.

Check the MPLS-TP check box in Create Cisco Router window.

Links and resource pools should now be visible in the MPLS-TP Details window, which is accessible from the **Inventory > Devices > MPLS-TP Details** page.

Verifying the MPLS-TP Discovery Results

After running MPLS-TP Discovery, you can see the result in various ways.

Viewing Logs

Once the **MPLS-TP Discovery** task is completed, you can view the log that is generated. This summary log will list any changes that have occurred in the MPLS-TP network. Discovery updates the logs with affected SR's in cases where the links in working or protect LSP no longer exist or have been changed. This could be as a result of node insertion/removal or simply changing a link number.

To view the log, select the relevant task in Task Manager and click **Logs**.

Verifying Links, Pools, and MPLS-TP Global and Router IDs

You can verify the status of links and pools by navigating to the MPLS-TP Details page at Inventory > Devices > MPLS-TP Details.

The MPLS-TP global and router IDs for a particular device can be verified by going to Inventory > Devices > Edit.

MPLS Label Sync

MPLS Label Sync task is to update the labels information. MPLS Labels can be out of sync due to manual provisioning. Hence, it is recommended to update the label information alone rather than the entire MPLS topology information often.

Similar to MPLS Discovery, MPLS Label Sync task can be performed from:

- Task Manager window
- Device Inventory window
- Device Creation window

MPLS-TP Labels sync task can also be done using this process.

Creating an MPLS-TP Policy

An MPLS-TP policy is needed to successfully create and deploy a service request. It serves as a template for the settings that are needed on the device.

To create an MPLS-TP policy, use the following steps:

Step 1 Choose one of the following:

a. **Service Design** > **Policy Manager**.

In the Policy Manager window, click **Create**.

b. **Service Design > Create Policy**.

In either case, a Policy Type drop-down appears.

Step 2 Click the down-arrow to open the **Policy Types** picker and select **MPLS-TP Tunnel**.

The Policy Information accordion opens.

Step 3 Complete accordion 1 – Policy Information.

Enter **Policy Name** and optionally a **Description**. Policy Name is the only field that is mandatory in the Policy Editor.

Step 4 Click **Next.**

The Policy Information accordion closes and the next accordion opens.

Step 5 Complete accordion 2 – Tunnel Characteristics.

Set how each of the attributes will be displayed within the Service Request Editor window using the drop-down next to each field:

- **Editable** will display the attribute and permit modification.
- **Visible** will display the attribute but prevent editing.
- **Hidden** will not display the attribute.

Make sure to select **Editable** for any fields that you want to be able to edit in the Service Request Editor.

Use the **State** field to indicate whether the tunnel should be provisioned with the **shutdown** command or not.

For path protection, ensure that the **Protection** box is selected so that Prime Provisioning auto generates an alternate protective path for the new tunnel.

For the **Diversity Options** drop-down menu, choose one of the following options:

- Node Diversity Required—Path calculation will fail if protection with unique nodes cannot be found.
- Node Diversity Desired—Allow a path with common nodes to be returned.
- Link Diversity Only—Do not allow working and protection path to pass through the same links.

Step 6 Complete accordion 3 – Tunnel End-points.

As in the previous accordion, remember to specify which fields should be Editable, Visible, and Hidden in the Service Request Editor.

Complete the fields as needed, using the drop-downs to select source and destination nodes and BFD templates.

Select the required BFD templates from the available list of BFD templates on the source and destination devices respectively or you can enter the BFD template name in the field irrespective of device or device type. A valid BFD template name is max. 31 characters long.

Note On IOS devices BFD timers are specified via a template, while on IOS-XR they are either specified globally and can be overridden in each individual TP tunnel. You can create a policy that works on IOS and IOS-XR, by definig the template and optionally by specifying timer values that would override the global BFD values in IOS-XR. During policy creation you are just providing default values, validation will be performed once you have actual values, when creating the individual tunnel service requests.

For an explanation of global ID and router ID, see Global ID and Router ID, page 7-9.

Step 7 Click **Finish** to create the policy.

The new policy appears in the list of tunnels in the Policy Manager.

Global ID and Router ID

Global ID and router ID are used to identify devices within the MPLS-TP network so they can be discovered and managed.

If you as a user decide to specify the router ID and global ID, those values will be used for tunnel creation. If they are not specified, the router ID and global ID configured on the device itself are used.

Every MPLS-TP tunnel and LSP has a unique ID formed by the concatenation of the Global ID, Router ID, Tunnel ID, and LSP ID of both ends of the tunnel. This ID is configured at every endpoint and midpoint of the tunnel. The Global ID and router ID are normally configured globally on a router but it is possible to override these values for specific tunnels. Prime Provisioning is aware of the globally configured IDs and uses them when configuring tunnels but also allows you to override these values as needed.

Global ID

Every MPLS-TP enabled node can have an MPLS-TP global ID configured within the global configuration. If the Global ID is set at the MPLS-TP global configuration level, it will be used as the default global ID for all endpoint and midpoint configuration. If not configured, a global ID of 0 is used for configured tunnels unless a different value is explicitly specified within the tunnel configuration itself.

The MPLS-TP global ID is retrieved from a device via MPLS-TP discovery.

Router ID

To be MPLS-TP enabled, a device must have a router ID.

If neither the MPLS-TP router ID nor the MPLS-TP global ID can be retrieved from the device, this is logged in the corresponding **MPLS-TP Discovery** task log file and all remaining MPLS-TP Discovery steps are halted for this device. The device in question is flagged as being MPLS-TP Disabled.

Creating an MPLS-TP Service Request

An MPLS-TP service request needs to be created to deploy a service request. It is assumed that at least one MPLS-TP policy is available. If not, see Creating an MPLS-TP Policy, page 7-7.

To create an MPLS-TP service request, use the following steps:

Step 1 This operation can be done in two ways:

a. From the Policy Manager, select the desired policy and click **Create Service Request**.

b. Choose **Operate > Create Service Request**.

The Service Request Editor window appears.

Next to the **Policy** field, click the down-arrow to open the policy picker.

Step 2 Select the desired MPLS-TP policy.

The Service Request Editor opens. In this editor,

Step 3 In the Service Request accordion, add a description in the **Service Description** field.

Step 4 In the Tunnel Characteristics accordion, use the pre-populated field values or make the desired modifications.

To set the **Diversity Options**, see Creating an MPLS-TP Policy, page 7-7 for an explanation.

Step 5 In the Tunnel End-Points accordion, complete the **Source Node** and **Destination Node** fields and optionally any other fields.

In this accordion, both source device, destination device, and BFD information is mandatory.

If the source device type is IOS, the BFD template details are mandatory or if the source device type is IOS-XR and the global template was not defined in the device, then you have to provide Source BFD min-interval and Source BFD Multiplier details.

Based on the chosen device type, the BFD template or the BFD attributes will be disabled accordingly as follows:

- If the device type is IOS, BFD attribute fields, such as BFD min-interval, BFD min-interval Standby, BFD Multiplier will be disabled.
- If the device type is IOS-XR, BFD template picker will be disabled.

If you have selected IOS/ IOS-XR device and entered the BFD template or the BFD attributes respectively, and then re-select IOS-XR/ IOS device respectively, the BFD template/ BFD attributes will be disabled without losing the values entered in the respective fields. So, if you try to re-select the device again, the BFD template/ BFD attributes will be enabled with the already entered values.

Note Validation of the BFD template against the device will be performed only for the IOS device as BFD template will be disabled for IOS-XR.

Step 6 In the Review Routing accordion, a default path is calculated and displayed automatically between source and destination.

Working path—Green solid line

Protect Link—Red dotted line

For an example of an MPLS-TP routing diagram, see Figure 7-3.

Figure 7-3 MPLS-TP Routing Diagram

- **Working Path Summary**—Click this button to view hop and link information for the working path.
- **Protect Path Summary**—Click this button to view hop and link information for the protect path.
- Add (or remove) path constraints by clicking the plus (or minus) icons to the right:
  - **Required NE/Link**—Specify network elements or links that traffic must pass through for either the working or the protect path.
  - **Excluded NE/Link**—Specify network elements or links that traffic must **not** pass through for either the working or the protect path.
- **Share common links between Working and Protect path**—Check this box to enable the common links in both Working and Protect Path.
  - To make use of common links or node in both Working and Protect Path add the common link or node as a **Required NE/Link** constraint in both working path and protect path.

For more information about path constraints, see Working with Path Constraints, page 7-12.

Step 7 Click **Calculate Path** to calculate the path.

Note In the case of Service Request modification the **Calculate Path** button will have the following options:

- **Working LSP**—Select this option to calculate and view the working path in the path diagram.
- **Protect LSP**—Select this option to calculate and view the protected path in the path diagram.
- **Both LSPs**—Select this option to calculate and view both the working and protected paths in the path diagram.

Step 8 Go back over the various accordions to check and edit as necessary.

Step 9 Click **Finish** on the last accordion to complete the create service request operation.

The Service Request Manager window opens.

Step 10 Click Deploy to deploy the service request at the time of creation itself.

Note During path calculation, the available and reserved bandwidth are not considered. So when a tunnel makes bandwidth reservation, the path with insufficient bandwidth could be considered. Sufficient bandwidth check occurs during 'Finish' or 'Deploy' operation. When there is insufficient bandwidth, the SR displays an error message with the corrective measures to be taken by the user. To make the path calculation pick a different path, use path constraints.

For information about the Service Request Manager elements and operations, see Chapter 10, "Managing Service Requests."

Guidelines for working with path constraints are provided in Working with Path Constraints, page 7-12.

A service request in the DRAFT state has not passed all validation and cannot be deployed. A service request in DRAFT state is marked by a white/orange work cone in the Service Request Manager.

[

Working with Path Constraints

Path constraints can be added to control the tunnel path when a service request is created or modified as shown in the procedure in Step 6 in the create procedure.

There are two ways to add path constraints:

- Clicking a node or link on the routing diagram and clicking the plus sign. This adds a new path constraint to the working path by default. Change to **Protect Path** using the drop down if needed. Similarly, clicking the minus sign will remove the constraint.
- If the node/link you want to exclude/include is not present in the diagram, you can use the selector next to **Required NE/Link.**

Note If you change anything after the first path calculation, for example adding/removing constraints, switching protection on/off, etc., you will need to re-run path calculation by clicking **Calculate Path.**

Running Config Audit

A config audit task can be run against an MPLS-TP service requests to check that the configuration rolled onto a device by a particular service request is still present as expected.

To create a MPLS-TP Config Audit task, use the following steps:

Step 1 Choose **Operate > Task Manager**.

Step 2 Click **Audit > Config Audit** to open the Create Task window.

Step 3 Modify the **Name** or **Description** fields as desired and click **Next**.

The service request selection window appears.

Step 4 Click **Select SRs** to add a service request and select schedule.

Step 5 Click **Submit**.

If successful, this adds the task to the list of created tasks in the Tasks window.

To view the task logs for the created tasks, in Task Manager select the created task and click **Logs**.

Running MPLS-TP Functional Audit

In an MPLS-TP Functional Audit, information is retrieved from source and destination endpoints to provide tunnel audit information.

This task only performs functional audit on service requests, which are not in one of the following states:

- **Draft**
- **Closed**
- **Requested**
- **Invalid**
- **Failed Deploy**

For more information on working with service requests, see Chapter 10, "Managing Service Requests."

To create a MPLS-TP Functional Audit task, use the following steps:

Step 1 Choose **Operate > Task Manager**.

Step 2 Click **Audit > MPLS-TP Tunnel Functional Audit** to open the Create Task window.

Step 3 Modify the **Name** or **Description** fields as desired and click **Next**.

The service request selection window appears.

Step 4 Click **Select SRs** to add a service request and select schedule.

Step 5 Click **Submit**.

If successful, this adds the task to the list of created tasks in the Tasks window.

To view the task logs for the created tasks, in Task Manager select the created task and click **Logs**.

Managing MPLS-TP Topology Changes

When a topology changes due to node insertion/removal, the MPLS-TP discovery has the ability to:

- Manage MPLS-TP topology change due to node insertion/removal.
- Identify MPLS-TP Tunnel SRs that has been impacted by node insertion/removal.
- Modify the impacted SRs to repair the MPLS-TP tunnels.
- Detect the MPLS-TP tunnel SRs that are affected by node insertion/removal.
- Re-calculate the path for affected SRs. During the recalculation:
  - Affected LSP is locked by Prime Network for uninterrupted traffic.

  - All affected SRs in Prime Provisioning are re-routed except those that are in Closed, Pending or In-Progress state or DELETE Op Type.
- Transition the affected SR into appropriate state.
  - Transition occurs only for deployed SRs.
  - If a new route is found for the broken tunnel for deployed SRs, the SR moves to Requested state.
  - A deployed tunnel SR moves into invalid state when no new route is found.
  - For all other SRs, except Closed, Pending or In-Progress State, and Op Type DELETE, the path is re-calculated without any state change.
- Report the affected SRs and update the SR logs. Discovery updates the logs with affected SRs in cases where the links in working or protect LSP no longer exist or have been changed.
  - For all the affected SRs, discovery updates the discovery logs and SR history report.
  - Discovery updates the SR history with:
  - Affected path, working/protect LSP.
  - State change details, previous/current state.
  - Messages related to path change/failure.
- Re-provision only the affected LSP. When the SR in Requested Modify state is selected to be deployed, only the LSP which has changed is re-provisioned by Prime Provisioning. This ensures that the traffic on the active LSP is uninterrupted.

Deploying an MPLS-TP Tunnel

The final step required to provision an MPLS-TP service request is the deploy the service request. This pushes the service request and the associated configuration updates to the network. Once the SR is successfully deployed, the bandwidth allocated by the SR is subtracted from the available bandwidth of each TP link used by the TP tunnel.

Note During bulk deployment, the bandwidth of the SR is reduced from the available bandwidth of the TP link for each successful deployment. Eventually, the available bandwidth is reduced to zero or less than the bandwidth requested by the consecutive SR. Such service request must be transitioned to invalid state due to insufficient bandwidth. Also, when the device does not have the enough bandwidth as requested by the SR, an insufficient bandwidth error message is displayed.

Tip A service request in **DRAFT** state cannot be deployed.

The deploy functionality is the same as for other Prime Provisioning services. For instructions on how to deploy an MPLS-TP service request, see Deploying Service Requests, page 10-9.

Decommissioning

MPLS-TP service request configurations can be removed from the network using the decommissioning functionality within the Service Request Manager. Decommissioning will cause the previously deployed configurations to be removed from all tunnel endpoint and mid-point devices within the MPLS-TP tunnel path.

Note Once the SR is successfully decommissioned and moved to CLOSED state, the bandwidth allocated to TP tunnel is added back to the available bandwidth of each TP link used by the tunnel.

To decommission one or more service requests, see Chapter 10, "Managing Service Requests."

Sample Configlets

The configlets included in this section show the CLIs generated by Prime Provisioning for particular services and features. Each configlet example provides the following information:

- Service
- Feature
- Devices configuration (network role, hardware platform, relationship of the devices and other relevant information)
- Sample configlets for each device in the configuration
- Comments.

All examples in this section assume the presence of an MPLS-TP core.

Note The configlets generated by Prime Provisioning are only the delta between what needs to be provisioned and what currently exists on the device. This means that if a relevant CLI is already on the device, it does not show up in the associated configlet.

This section provides sample configlets for MPLS-TP service provisioning in Prime Provisioning.

It includes the following section:

MPLS-TP Working Tunnel Configlet (IOS)

Configuration

- Service: MPLS-TP Working Tunnel
- Feature: MPLS-TP configlet (IOS) for configuring MPLS-TP enabled nodes.

Configlets

IOS Device Configuration	Comments

Endpoint Config
`interface Tunnel-tp200`
`  description PrimeF:JobID:2(testTunnel)`
`  tp tunnel-name test`
`  tp bandwidth 100`
`  tp source 3.3.3.3 global-id 2`
`  tp destination 1.1.1.1 tunnel-tp 200 global-id 3`
`  bfd BFDTemplate-SingleHopMicrosec-1`
`  working-lsp`
`  lsp-number 0`
`  in-label 8018`
`  out-label 5003 out-link 8`
`  protect-lsp`
`  lsp-number 1`
`  in-label 8019`
`  out-label 50012 out-link 12`
 	
Midpoint Config
`mpls tp lsp source 3.3.3.3 global-id 2 tunnel-tp 200 lsp working destination 1.1.1.1 global-id 3 tunnel-tp 200`
`  forward-lsp`
`  tp bandwidth 100`
`  in-label 5003 out-label 50011 out-link 10`
`  reverse-lsp`
`  tp bandwidth 100`
`  in-label 5004 out-label 8018 out-link 8`
 	
EndPoint Config
`interface Tunnel-tp200`
`  description PrimeF:JobID:2(testTunnel)`
`  tp tunnel-name test`
`  tp bandwidth 100`
`  tp source 1.1.1.1 global-id 3`
`  tp destination 3.3.3.3 tunnel-tp 200 global-id 2`
`  bfd BFDTemplate-SingleHopMicrosec-1`
`  working-lsp`
`  lsp-number 0`
`  in-label 50011`
`  out-label 5004 out-link 10`
`  protect-lsp`
`  lsp-number 1`
`  in-label 50012`
`  out-label 8019 out-link 12`
	`Create an MPLS-TP working tunnel with endpoint and midpoint nodes. This involves configuring the settings on each node in the tunnel.`
	
	`Create an MPLS-TP working tunnel with the following attributes:`
	
	`Endpoint 1:`
	`- tp tunnel name: test`
	`- Source: 3.3.3.3`
	`- Destination 1.1.1.1`
	`- Bandwidth 100 kbps`
	`- bfd BFDTemplate-SingleHopMicrosec-1`
	`- Working LSP configuration`
	`- Protect LSP configuration`
	
	`Midpoint:`
	`- Source: 3.3.3.3`
	`- Destination 1.1.1.1`
	`- Bandwidth 100 kbps`
	`- Forward LSP configuration`
	`- Reverse LSP configuration`
	
	`Endpoint 2:`
	`- tp tunnel name: test`
	`- Source: 1.1.1.1`
	`- Destination 3.3.3.3`
	`- Bandwidth 100 kbps`
	`- bfd BFDTemplate-SingleHopMicrosec-1`
	`- Working LSP configuration`
	`- Protect LSP configuration`

MPLS-TP Working Tunnel Configlet (IOS-XR)

Configuration

- Service: MPLS-TP Working Tunnel
- Feature: MPLS-TP configlet (IOS-XR) for configuring MPLS-TP enabled nodes.

Configlets

IOS-XR Device Configuration	Comments

Endpoint Config
`interface tunnel-tp0`
`  description PrimeF:JobID:2 (testTunnel)`
`  source 3.3.3.3`
`  destination 1.1.1.1 global-id 8 tunnel-id 1`
`  working-lsp`
`  in-label 36`
`  out-label 23 out-link 12`
`  lsp-number 0`
`  protect-lsp`
`  in-label 37`
`  out-label 33 out-link 100`
`  lsp-number 1`
`  bfd`
`  min-interval 50`
`  min-interval standby 50`
`  multiplier 3`
 	
Midpoint Config
`mpls traffic-eng tp`
`  mid 3.3.3.3_1_protect_3.3.3.4_0`
`  source 3.3.3.3 tunnel-id 1 global-id 8`
`  destination 1.1.1.1 tunnel-id 0 global-id 80`
`  forward-lsp`
`  in-label 32 out-label 37 out-link 100`
`  reverse-lsp`
`  in-label 33 out-label 24 out-link 10`
 	
EndPoint Config
`interface tunnel-tp1`
`  description PrimeF:JobID:2(testTunnel)`
`  source 1.1.1.1`
`  destination 3.3.3.3 global-id 80 tunnel-id 0`
`  working-lsp`
`  in-label 23`
`  out-label 36 out-link 4`
`  lsp-number 0`
`  protect-lsp`
`  in-label 24`
`  out-label 32 out-link 10`
`  lsp-number 1`
`  bfd`
`  min-interval 50`
`  min-interval standby 50`
`  multiplier 3`
	Create an MPLS-TP working tunnel with endpoint and midpoint nodes. This involves configuring the settings on each node in the tunnel.
	
	Create an MPLS-TP working tunnel with the following attributes:
	
	Endpoint 1:
	- Source: 3.3.3.3
	- Destination 1.1.1.1
	- Bandwidth 100 kbps
	- Working LSP configuration
	- Protect LSP configuration
	
	Midpoint:
	- Source: 3.3.3.3
	- Destination 1.1.1.1
	- Bandwidth 100 kbps
	- Forward LSP configuration
	- Reverse LSP configuration
	
	Endpoint 2:
	- Source: 1.1.1.1
	- Destination 3.3.3.3
	- Bandwidth 100 kbps
	- Working LSP configuration
	- Protect LSP configuration