CHAPTER 34

Troubleshooting MPLS VPNs

This chapter provides information about troubleshooting MPLS VPNs.

General Troubleshooting Guidelines

For general troubleshooting of failed provisioning, perform the following steps.

Step 1 Identify the failed service request and go into **Details**.

a. To do this, go to the Service Request Editor and click **Details**.

Of main concern is the status message—this tells you exactly what happened.

b. If the status message tells you it's a failed audit, click the **Audit** button to find out exactly what part of the audit failed.

Step 2 If the troubleshooting sequence in Step 1 does not give you a clear idea as to what happened, use the logs in the Task Manager to identify the problem.

a. To do this, choose **Monitoring** > **Task Manager** > **Logs** > **Task Name**.

b. There is a lot of information in this log. To isolate the problem, you can use the filter. If you filter by log level and/or component, you can usually reduce the amount of irrelevant information and focus on the information you must know to locate the problem.

Step 3 Also see the section Frequently Asked Questions, page 34-2 in this appendix for information on some common questions and issues.

Gathering Logs for Development Engineering

Go through the troubleshooting steps described in General Troubleshooting Guidelines, page 34-1. If you have failed to troubleshoot or identify the problem, this section provides information on how to gather logs for the development engineer to troubleshoot.

Tip The logs apply to both MPLS VPNs and Layer 2 VPNs.

There is a property in DCPL called **Provisioning.Service.mpls.saveDebugData**. If this property is set to **True**, whenever a service request is deployed, a temporary directory is created in PRIMEF_HOME/tmp/mpls.

The directory contains the job ID of the service request prefixed to it, along with a time stamp. This directory contains the uploaded configuration files, service parameters in XML format, and the provisioning and audit results.

The default is set to True.

To verify, perform the following steps.

Step 1 Locate the property by choosing **Administration** > **Control Center**.

The Control Center Hosts page is displayed.

Step 2 Check the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Click **Config**.

The Host Configuration window is displayed.

Step 4 Navigate to **Provisioning** > **mpls**.

Step 5 Click **saveDebugData**.

Frequently Asked Questions

Below is a list of FAQs concerning MPLS VPN provisioning.

What is the MPLS provisioning workflow?

The tasks listed below depict the MPLS provisioning workflow. This section assumes an operator deploys a service request using a caller such as Task Manager.

1. The Provisioning driver (ProvDrv) gets the service request to be deployed.
2. From the service request, the Provisioning driver deduces which devices are involved.
3. The latest router configurations must be obtained, so the Provisioning driver tells the Generic Transport Library (GTL)/ Device Configuration Service (DCS) to upload the latest router configurations. The result is used by the service module.
4. The Provisioning driver determines what service modules are involved based on the service and device types.
5. The Provisioning driver queries the Repository for the service intention. The Provisioning driver sends the service intention to the service module, along with the uploaded configuration.
6. The service module generates configlets based on the configurations and service intention and returns the appropriate configlets to the Provisioning driver.
7. The Provisioning driver signals GTL/DCS to download the configlets to the target routers.
8. The Provisioning driver sends the updated result, including the download result, to the Repository, which then updates its state.

Definitions of terms mentioned in the above steps.

- **Device Configuration Service (DCS)**: Responsible for uploading and downloading configuration files.
- **Generic Transport Library (GTL)**: Provides APIs for downloading configlets to target devices, uploading configuration files from target devices, executing commands on target devices, and reloading the target device.

  This library provides a layer between the transport provider (DCS) and the client application (for example, the Provisioning Driver, Auditor, Collect Config operation, Exec command). The main role of the GTL is to collect the target specific information from the Repositories and the *properties* file and pass it on to the transport provider (DCS).
- **ProvDrv (the Provisioning driver)**: ProvDrv is the task responsible for deploying one or more services on multiple devices.

  ProvDrv performs the tasks that are common to all services, such as the just-in-time upload of configuration files from the devices, invocation of the Data Driven Provisioning (DDP) engine, obtaining the generated configlets or the audit reports from the DDP engine, and downloading the configlets to the devices.
- **Repository**: The Repository houses various IP Solution Center data. The Prime Fulfillment Repository uses Sybase or Oracle.
- **Service module**: Generates configlets based on the service types.

What do I do if my task does not execute even if I schedule it for immediate deployment?

This problem is likely due to one of the Prime Fulfillment servers being stopped or disabled.

To check the status of all Prime Fulfillment servers, perform the following steps.

Step 1 Open the Host Configuration dialog by going to **Administration** > **Control Center**.

The Control Center Hosts page is displayed.

Step 2 Check the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Choose **Servers**.

The Server Status page appears, as shown in Figure 34-1.

Figure 34-1 Prime Fulfillment Server Status

Step 4 On the Prime Fulfillment server, use the **wdclient status** command to find out the detailed status of the server.

What do I do when a service request is in the Wait Deployed state?

This concerns the devices that are configured to use Cisco Configuration Engine as the access method. If the devices are offline and a configlet was generated for it, the service request will move into the Wait Deployed state. As soon as the devices come online, the list of configlets will be downloaded and the status of the device will change.

What do I do when a service request is in the Failed Audit state?

At least one command is missing on the device. Perform the following steps.

Step 1 From the Prime Fulfillment user interface, go to **Service Request Editor** > **Audit** > **Audit Config**.

Step 2 Check the list of commands that are missing for each device.

Step 3 Look for any missing command that has an attribute with a default value.

What do I do if the service request is in the same state as it was before a deployment?

If after a deployment a service request state remains in its previously nondeployed state (Request, Invalid, or Pending), it's an indication that the provisioning task did not complete successfully. Use the steps described in General Troubleshooting Guidelines, page 34-1 to find out the reason for the service request failure.

What do I do if I receive the following out-of-memory error: OutOfMemoryError?

Perform the following steps.

Step 1 Open the Host Configuration dialog by choosing **Administration** > **Control Center**.

The Control Center Hosts page is displayed.

Step 2 Check the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Click **Config**.

The Host Configuration window is displayed.

Step 4 Navigate to **watchdog** > **servers** > **worker** > **java** > **flags**.

Step 5 Change the following attribute:

Change the **Xmx256M** attribute to **Xmx384M** or **Xmx512M**.

What do I do if Prime Fulfillment will not remove a route target import/export for a VPN?

Scenario: When an MPLS service request is edited to be associated to a new VPN, the old VPN will only be removed if it is associated with only one interface. The relationship between the service request and the customer is via the VPN. The optional Customer field in a service request does not have any bearing on configuration. For example, if an MPLS service request for *custA* exists with *vpnB/cercB*, but needs to be modified to reflect *vpnA/cercA*, modifying the service request to use *vpnA/cercA* will not remove the route target for *vpnB* from the *vrfB* if there is more than one interface associated with the same VRF.

Recommended Action Running the same scenario with only one interface referring to *vrfB*, Prime Fulfillment will remove *vrfB* and correctly add *vrfA* with route target *A*.

Why does my service request go to Invalid when I choose provisioning of an extra CE Loopback interface?

It is possible that the auto pick option of the IP addresses was selected for the service request, but a /32 IP address pool was not defined. Check and make sure the IP address and the IP address pool defined for this service request are compatible.

When saving a service request, why does it say "CERC not initialized"?

It is necessary to pick a CERC for the link to join. Please check the service request to see if a CERC was selected.

Why does creation of a VLAN ID pool require an Access Domain?

VLAN ID pools are associated with an Access Domain. Access Domains model a bridged domain; VLAN IDs should be unique across a Bridged Domain.

PE-POPs must be associated with an Access Domain. An Access Domain can have more than one PE-POP associated with it.

In a Paging table, why are the Edit and Delete options disabled, even though only one check box is checked?

This is possible if one or more check boxes are selected in previous windows.

Why can I not edit an MPLS VPN or L2VPN policy?

If a service request is associated with a policy, that policy can no longer be edited.

I am unable to create a CERC—can you explain why?

You have to define a Route Target pool before you create a CERC, unless you specify the Route Targets manually.

How can I modify the configlet download order between the PE, CE, and PE-CLE devices?

There is a property called **Provisioning.Services.mpls.DownloadWeights.*** that allows you to specify the download order for the following device types: PE, CE, PE-CLE, and MVRF CE.

For example, to ensure that the configlet is downloaded to the PE before it is downloaded to the CE, configure the **Provisioning.Services.mpls.DownloadWeights.weightForPE** property with a weight value greater than that of the CE.

What does the property Provisioning.Service.mpls.reapplyIpAddress do?

If this property is set to True, during deployment of a decommissioned service request, this property will keep the IP address on the CE and PE intact on the router to maintain IPv4 connectivity to the CE.

When I create a multi-hop NPC between a CE and PE through at least one PE-CLE device, why do I see some extra NPCs created?

IP Solution Center creates the extra NPCs to prevent operators from having to enter the same information again. A CE can now be connected to the PE-CLE device, and a new NPC will be created that will connect the new CE to a PE over the PE-CLE-to-PE NPC link.

During service request provisioning, in the Interface selection list box, why don't I see the entire list of interfaces on the device?

This is probably due to a particular interface type being specified in the service policy. If that is the case, only interfaces of the specified interface type are displayed.

Why does my service request go to Invalid with the message "loopback address missing"?

This is a Layer 2 VPN question.

This is because the loopback address required to peer the pseudowire between PEs has not been defined in the PE-POP object in Prime Fulfillment.

What is the intent of the Allocate New Route Distinguisher check box in the MPLS policy?

There were some behavior changes implemented in Prime Fulfillment that differ from the legacy product "VPNSC". In VPNSC, VRFs were PE centric. Therefore, the behavior was for a new VRF to be configured for each VPN on a PE router. This behavior was modified in Prime Fulfillment to make VRFs VPN centric. For most of routing, the VRF/route distinguisher (RD) is only PE significant, except when doing iBGP load balancing. For this reason, it is possible to use the same values for a single VPN on all PE routers. This is more convenient for the user in context of troubleshooting, reporting, etc.

To increase flexibility for users where there is iBGP load balancing and also to address custom solutions and needs, there are two options available in Prime Fulfillment. One is VRF and RD Overwrite, and the other is Allocate New Route Distinguisher. VRF and RD Overwrite is exactly like it sounds. This gives the user the ability to force the VRF name and RD values for a link being provisioned. This is useful for joining a pre-existing VRF that was not provisioned by Prime Fulfillment.

Note Once you specify values to sub-attributes under the VRF and RD Overwrite attribute (that is, the VRF Name and RD Value attributes) and save an MPLS service request, both of these fields are disabled and are no longer editable. This behavior was introduced because changing the default values for the VRF Name and RD Value can alter or disable currently running service requests. Therefore, if these values need to be changed on a deployed service request, the workaround is that you must decommission and purge the service request and create a new service request. In the case of a new service request that has not yet been deployed, you must force purge the service request and then create a new service with new values.

The second option, Allocate New Route Distinguisher, is only valid for configuring a new VRF and RD on a PE router for the first time. This mimics the VPNSC behavior of individual VRFs per PE router. The following is the rule for new RD when a pre-existing VPNSC repository is not involved:

When Allocate New Route Distinguisher is enabled:

- Create a new VRF if there is no matching VRF configuration on that PE.
- If there is matching VRF configuration on that PE, then reuse it.

When Allocate New Route Distinguished is disabled:

- Find the first matching VRF configuration across the whole range of PEs, regardless of the PE, if this VRF is found on the PE being configured, reuse it. If it is not found on the PE create it.
- Note: The service request might get a VRF that has already been configured on another PE router.

An issue with pre-existing VRFs that were configured under VPNSC is that in VPNSC the Allocate New Route Distinguisher flag was always turned on. Thus, when you apply the flag again, Prime Fulfillment first looks for an existing VRF on the PE. It uses that VRF (in this case, the one provisioned by VPNSC). If no VRF is found, Prime Fulfillment creates a new VRF. When adding a new link to old VPNSC links, if the Allocate New Route Distinguisher flag is not turned on, Prime Fulfillment finds the first matching VRF configured across the network. If the PE does not have this VRF, Prime Fulfillment will create it on the router.

Use cases:

1. When adding a link to an existing PE with a legacy (VPNSC) VRF, you must select the Allocate New Route Distinguisher option.
2. When adding a link to a new PE, if you desire VRF/RD values that have not been configured before in this VPN, then you must select the Allocate New Route Distinguisher option.
3. When adding a new link to a new PE, if you want to reuse a VRF/RD value that has been used elsewhere in the network, then you must select the VRF and RD Overwrite option.
4. If you provisioned a link that has incorrect VRF/RD values (that is, not matching those previously provisioned by VPNSC), the link will need to be modified and redeployed. During the modification, you must select the VRF and RD Overwrite option and specify the same VRF/RD values used in VPNSC.
5. If you are planning to deploy iBGP load balancing across multiple PEs, the Allocate New Route Distinguisher option should be always enabled. This is to make sure the condition for unique RD is met, in order to satisfy load balancing requirements.

How can an MPLS service request using standard UNI ports allow CDP packets?

By default, an MPLS service request creates MAC ACLs for a standard UNI that restricts access of BPDU handling on the Layer2 control plane. The created ACLs are similar to the following:

```
interface FastEtherent0/15
mac access-group ISC-$name in
mac access-list extended ISC-$name

deny any host 0180.c200.0000 ===> PVST, MSTP, RSTP, and STP
deny any host 0100.0ccc.cccd ===> PVST+
deny any host 0100.0ccc.cccc ===> CDP, VTP, DTP, UDLD, PAgP
deny any host 0100.0ccd.cdd0 ===> CDP,VTP,STP
permit any any
```

Note The text appearing after “===>” is not part of the MAC ACL. It is a list of which protocols are blocked by each MAC address.

Alternatively, when the MPLS service request is created, you can edit the link attributes and perform the following steps.

Step 1 Enable **Use Existing ACL Name**.

This will enable the Port-Based ACL Name option

Step 2 Enter an empty or non-existing MAC ACL name.

When the MPLS service request is deployed, it will no longer issue the default BPDU filtering MAC ACLs. Instead, it will create an **access-group** command on the UNI interface that points to an empty ACL. Example:

```
interface FastEthernet0/15 mac access-group {$PACL_NAME} in
```

No MAC ACL is created.

Is it possible to use 2 or 3 address pools when creating an L3 VPN?

Imagine that you have IP pool 10.10.10.0/24 assigned to a region, and a PE is assigned to this region. What if one customer is using the same subnet in his LAN range? This forces you to use another subnet for the PE-CE link. How is this handled by Prime Fulfillment? The only way is to do it manually, without using auto pick. Prime Fulfillment does not support for the use of different address pools for different customers.

Another related issue is as follows. If a customer is using the same IP addresses inside his LAN segment as are used in the Prime Fulfillment pool of IP addresses, this causes a problem. For this reason, you must have multiple subnets for the PE-CE IP addresses, and use the suitable one (one that does not conflict with the IP addresses used by the customer). When you create an IP address pool, the repository knows the range, and will not allow you to use overlapping IP addresses as part of the pool.
Prime Fulfillment does not have any support for different pools to be used within the same PE.
Prime Fulfillment allows you to create multiple pools, but you can only use one based on the provider region. Prime Fulfillment picks up the next in line if the first pool runs out of IP addresses. There is no selection mechanism for you to select which pool will be used with auto pick. You can use manually added IP addresses, as long as the IP address do not overlap with the pool.

When will an IP address from the MPLS IP address pool be returned to the available pool after the service request is decommissioned?

When a service request is decommissioned, the IP address is returned back to the available pool after the service request goes to the DEPLOYED state. Prime Fulfillment prevents reuse of the returned IP addresses by a new service request for about twenty-four hours. The same behavior applies when the service request is decommissioned and then purged.

Why doesn't Prime Fulfillment remove some of the router BGP/EIGRP commands when a service request is decommissioned?

Prime Fulfillment removes the address family CLIs from router BGP or EIGRP configurations if and only if the VRF is removed. For router EIGRP, the process is not removed due to the potential presence of other CLIs that were not configured by Prime Fulfillment. This is particularly applicable when the network statement was added outside of Prime Fulfillment. Prime Fulfillment does not remove the redistribution from other routing protocols under EIGRP because the redistribute command might not be created specifically for the link.

Prime Fulfillment only removes the router OSPF process if the VRF is removed. This applies only for a PE. For a CE, router OSPF is removed if the network statement is removed. Prime Fulfillment does not remove router BGP nor router EIGRP.

What happens if the platform or IOS (or IOS XR) version does not support Q-in-Q (for example WS-X6724-SFP)?

The service request will result in a **Failed Deploy** state, and the log file will be similar to the following

For IOS:

```
SEVERE Provisioning.ProvDrvDownload failed for device NPE-1: 315 : Error downloading
cmd=[encapsulation dot1Q 158 second-dot1q 1510], response=[encapsulation dot1Q 158
second-dot1q 1510^
% Invalid input detected at '^' marker.NPE-1(config-subif)#]
```

For IOS XR:

```
SEVERE Provisioning.ProvDrvDownload failed for device NPE-1: 315 : Error downloading
cmd=[encapsulation dot1Q 158 1510], response=[encapsulation dot1Q 158 1510^
% Invalid input detected at '^' marker.NPE-1(config-subif)#]
```

Edit the service request, disable second VLAN ID, and then re-deploy.

Why doesn’t Prime Fulfillment provision Q-in-Q , although the hardware/IOS does support Q-in-Q?

Possible errors:

- The port is in switchport mode. Solution: Check the port configuration, and if necessary, run **no switchport**.
- The SVI flag is enabled. Solution: Disable SVI.

Why does a port with existing subinterfaces (Q-in-Q) plus SVI on same interface result in INVALID?

If you modify a service request with only one sub interface to SVI enabled, then the service request goes to the Deployed state (in the case of an IOS device). If you create a new service request with the same interface (that is, an existing subinterface) with SVI enabled, the service request goes to the Invalid state.

Is it possible to deploy single dot1q and Q-in-Q service requests under the same interface/port?

Yes.

How can I remove the second VLAN ID from a service request that is Deployed with Q-in-Q?

You must edit/modify the service request, remove the second VLAN ID entry, and redeploy the service request. A configlet like the following will be created:

```
interface GigabitEthernet2/0/15.158
no encapsulation dot1Q
encapsulation dot1Q 158
ip address 10.1.1.105 255.255.255.252
```