Troubleshooting MPLS VPN

This chapter contains the following major sections:

MPLS VPN 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.

Terms Defined

- **Device Configuration Service (DCS)**: Responsible for uploading and downloading configuration files.
- **Generic Transport Library**: 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 ISC Repository uses Sybase or Oracle.
- **Service module**: Generates configlets based on the service types.

General Troubleshooting Guidelines

For general troubleshooting of failed provisioning, follow these 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 doesn't 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.

Common Provisioning Issues

Below is a list of common provisioning problems and recommended solutions.

Symptom 1

My task does not execute even if I schedule it for immediate deployment.

Recommended Action

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

Step 1 To check the status of all ISC servers, open the Host Configuration dialog by choosing **Administration > Control Center**.

The Control Center Hosts page is displayed.

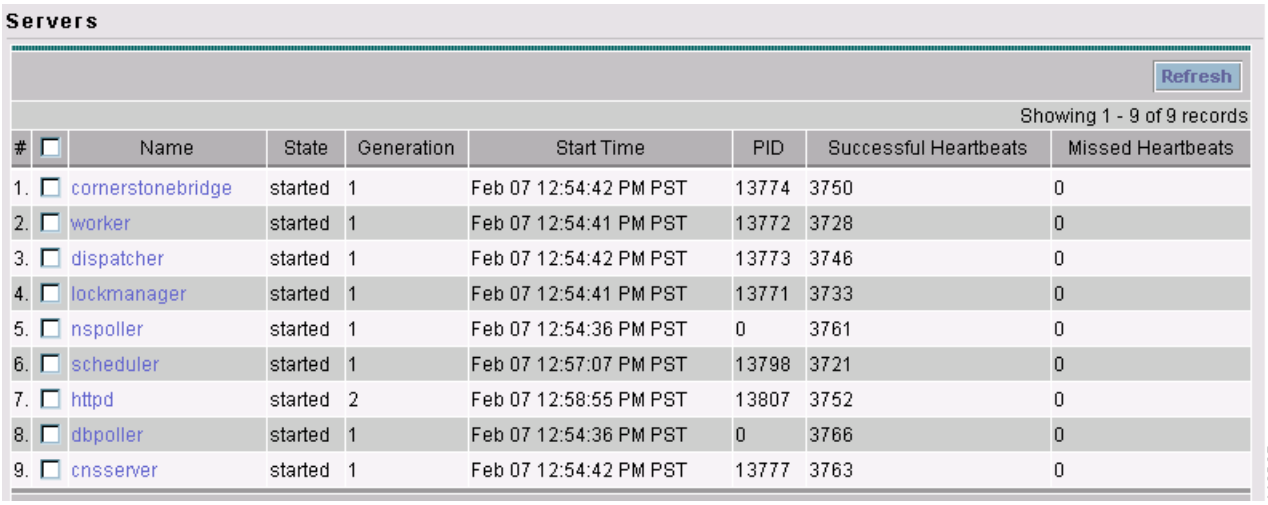
Step 2 Click the check box for the host of interest.

The menu buttons for the Hosts page are enabled.

Step 3 Select **Servers**.

The Server Status page is displayed (see Figure C-1).

Figure C-1 ISC Server Status



Servers

Refresh

Showing 1 - 9 of 9 records

#	Name	State	Generation	Start Time	PID	Successful Heartbeats	Missed Heartbeats
1.	cornerstonebridge	started	1	Feb 07 12:54:42 PM PST	13774	3750	0
2.	worker	started	1	Feb 07 12:54:41 PM PST	13772	3728	0
3.	dispatcher	started	1	Feb 07 12:54:42 PM PST	13773	3746	0
4.	lockmanager	started	1	Feb 07 12:54:41 PM PST	13771	3733	0
5.	nspoller	started	1	Feb 07 12:54:36 PM PST	0	3761	0
6.	scheduler	started	1	Feb 07 12:57:07 PM PST	13798	3721	0
7.	httpd	started	2	Feb 07 12:58:55 PM PST	13807	3752	0
8.	dbpoller	started	1	Feb 07 12:54:36 PM PST	0	3766	0
9.	cnsserver	started	1	Feb 07 12:54:42 PM PST	13777	3763	0

116287

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

Symptom 2

The service request is in the Wait Deployed state.

Recommended Action

This concerns the devices that are configured to use the CNS 2100 Series Intelligence 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.

Symptom 3

The service request is in the Failed Audit state.

Recommended Action

At least one command is missing on the device.

Step 1 From the ISC 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.

Symptom 4

The service request is in the same state as it was before a deployment.

Recommended Action

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 C-2 to find out the reason for the service request failure.

Symptom 5

You receive the following out-of-memory error: *OutOfMemoryError*.

Recommended Action

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

The Control Center Hosts page is displayed.

Step 2 Click 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**.

Troubleshooting MPLS VPN and Layer 2 VPN

Go through the troubleshooting steps described in General Troubleshooting Guidelines, page C-2. If you have failed to troubleshoot or identify the problem, the information in 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 *ISC_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.

Step 1 To verify, you can locate the property by choosing **Administration > Control Center**.

The Control Center Hosts page is displayed.

Step 2 Click 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. (Question 13 pertains to Layer 2 VPNs.)

Q 1: Why does my service request go to Invalid when I select 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.

Q 2: 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.

Q 3: 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.

Q 4: 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.

Q 5: 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.

Q 6: I’m 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.

Q 7: 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's downloaded to the CE, configure the **Provisioning.Services.mpls.DownloadWeights.weightForPE** property with a weight value greater than that of the CE.

Q 8: What does this 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.

Q 9: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.

Q 10: 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.

Q 11: Why do BGP and EIGRP not appear in the Routing protocol selection list for a service request associated to a No-CE policy?

BGP and EIGRP require certain CE-related parameters, such as the customer AS number and the CE's IP address. Since none of these parameters are requested in a No-CE policy, it is not feasible to provision these protocols. To provision a service request with BGP or EIGRP, use a policy with the **CE present** option specified, and you can set the CE to **unmanaged**.

Q 12: Why do the routing protocols BGP and EIGRP not appear when I select **No CE**?

If there is no CE in the scenario, BGP and EIGRP are not supported.

Q 13: This is a Layer 2 VPN question: Why does my service request go to Invalid with the message "loopback address missing"?

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

Troubleshooting IPsec Mapping into MPLS

IPsec mapping into MPLS consists of an IPsec service request and an MPLS service request. Each has its own debugging mechanism. There is no common debugging methodology for both IPsec and MPLS since they are two independent service requests. **- IPsec is not supported in this release. -**