Prime Cable Provisioning Components and Deployment

The deployment of Prime Cable Provisioning in a broadband service-provider network involves the deployment of the various components associated with the product. These components can then be configured to manage the network.

Components of Prime Cable Provisioning

This section describes the basic Prime Cable Provisioning components, such as:

Regional Distribution Unit (RDU) that provides:

- The authoritative data store of the Prime Cable Provisioning system.
- Support for processing application programming interface (API) requests.
- Monitoring of the system's overall status and health.

- RBAC for better user management.

See Regional Distribution Unit, on page 3 for additional information.

Provisioning Web Services (PWS) that provides:

- A SOAP based web service for device provisioning functions.

- Support for both HTTP and HTTPS connectivity.

- Supports interacting with multiple RDU servers.

See Provisioning Web Service, on page 31 for additional information.

Device Provisioning Engines (DPEs) that provide:

- Interface with customer premises equipment (CPE).

- Configuration cache.

- Autonomous operation from the RDU and other DPEs.

- PacketCable provisioning services.

- Dual-stack provisioning

- IOS-like command-line interface (CLI) for configuration.

See Configuring Device Provisioning Engines and Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide for additional information.

Prime Cable Provisioning API that provides total client control over system capabilities

See Cisco Prime Cable Provisioning 5.3 Integration Developers Guide for additional information about the APIs.

Cisco Prime Network Registrar servers that provide:

- Dynamic Host Configuration Protocol (DHCP).

- Domain Name System (DNS).

See Cisco Prime Network Registrar, on page 41 for additional information.

Provisioning Groups that provide:

- Logical grouping of Network Registrar servers and DPEs in a redundant cluster.

- Redundancy and scalability.

See Provisioning Groups, on page 45 for additional information.

A Kerberos server (KDC) that authenticates PacketCable Multimedia Terminal Adapters (MTAs). **See** Key Distribution Center, on page 43, **for additional information**.

The Prime Cable Provisioning process watchdog that provides:

- Administrative monitoring of all critical Prime Cable Provisioning processes.

- Automated process-restart capability.

- Ability to start and stop Prime Cable Provisioning component processes.

See Prime Cable Provisioning Process Watchdog for additional information.

An SNMP agent that provides:

- Third-party management systems.

- SNMP version v2.

- SNMP Notification.

See SNMP Agent for additional information.

An Admin UI that supports:

- Adding, deleting, modifying, and searching for devices.

- Configuring of global defaults and defining of custom properties.

- Configuring groups.

- Configuring servers and Provisioning Groups.

- Configuring RBAC.

See Administrator User Interface, on page 44 for additional information.

Regional Distribution Unit

The RDU is the primary server in the provisioning system. You must install the RDU on a 64-bit server running either Solaris or Linux operating systems.

The functions of the RDU include:

- Managing device configuration generation

- Generating configurations for devices and distributing them to DPEs for caching

- Synchronizing with DPEs to keep device configurations up to date

- Processing API requests for all Prime Cable Provisioning functions

- Managing the Prime Cable Provisioning system

The RDU supports the addition of new technologies and services through an extensible architecture.

Prime Cable Provisioning supports one RDU per installation. You could configure high availability for the RDU. To provide failover support, we recommend using the clustering software from Symantec or Oracle. We also recommend using RAID (Redundant Array of Independent Disks) shared storage in such a setup. RDU also supports Global Server Load Balancing (GSLB) to enable failover support and continue the RDU service in case the primary RDU service fails.

The following sections describe these RDU concepts:

- Authentication Support, on page 28

- GSLB Support, on page 31

- Configuration Regeneration Service (CRS), on page 24

High Availability for RDU

RDU as the backbone of Prime Cable Provisioning must be fault tolerant and reliable. An RDU crash can cause severe data losses resulting in discontinuity of the cable provisioning service. RDU can crash for any of the following reasons:

- Electrical outage

- Network outage due to network malfunctioning

- Overheating of server

- Operating System crash

- Hard-disk failure

- RDU process becomes unresponsive

- Database corruption

- Database corruption due to incomplete transaction

- Malfunctioning of infrastructure software

- Race condition between primary and secondary RDU

To avoid this, Prime Cable Provisioning provides RDU High Availability (RDU HA) on Linux operating system RHEL 6.5 and CentOS 6.5 (both 64-bit). High Availability (HA) is the duplication of critical components or functions of a system with the intention of increasing reliability of the system. HA is assured by making a redundant pair, which will fail over to a peer in case of any outage or service breakdown. The redundant pair is referred to as primary and secondary RDU nodes in Prime Cable Provisioning.

RDU HA clustering is an active-passive setup (1:1 node setup), which means that only one active RDU will function at any given time. RDU HA ensures the following:

- Virtual IP (VIP) based switching between primary and secondary RDU node.

- Failover between primary and secondary node.

- Database replication between primary and secondary by means of block level synchronization.

- Prime Cable Provisioning configuration files replication between primary and secondary RDU nodes. These are RDU configuration files.

- Provisioning of manual and automatic failback.

- Recovery of corrupted (impacted) node from active RDU, without disturbing the active RDU.

You can also perform the installation to make it HA Ready initially, and later configure the HA cluster, when required, using a special installation mode, *Configure HA*.

For more information about configuring, monitoring and troubleshooting RDU redundancy, see Scripts to Manage and Troubleshoot RDU Redundancy.

For installation, configuration and deployment details, see the Cisco Prime Cable Provisioning 5.3 Quick Start Guide.

Figure 1: RDU Redundancy
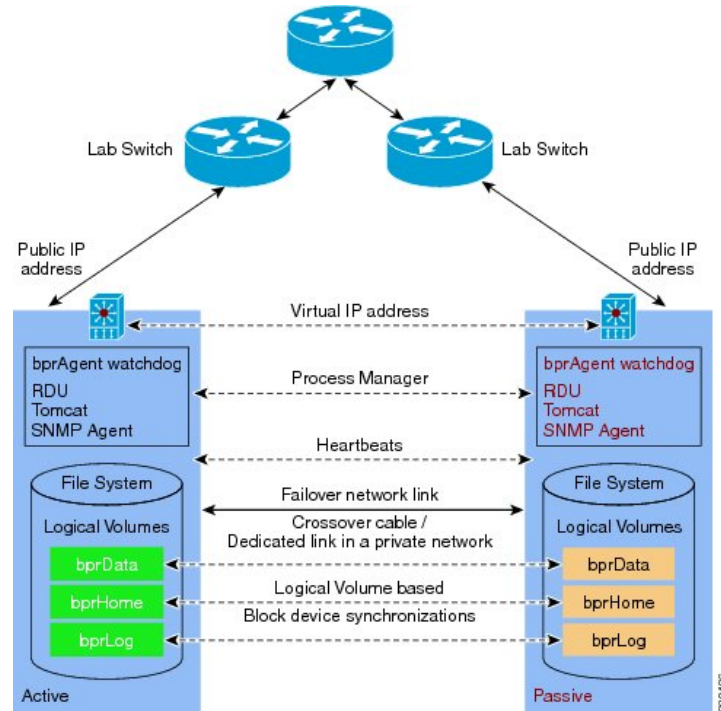
The solid line indicates physical network connectivity between the nodes. The dotted lines indicate logical synchronization between the two nodes.

1:1 Active-Passive Setup

A high performance system dedicates one secondary for each primary, a 1:1 failover relationship, where the secondary is an exact replica of the primary, including configuration information. To ensure RDU redundancy, the following active-passive setup is established:

- By default, Prime Cable Provisioning resources including the VIP resource are active only on the primary node. In case of a failover, they are migrated to the available active node (secondary node).

- Failure of the HA configured resources on active can lead in to a failover to peer (passive) node.

- Automatic failover and manual failback support.

- Minimized race-conditions and split-brain situations.

- Configurable failure stand-by timeout for automatic failback. Automatic failback starts when the CRM resources are cleaned up on the failed node.

Figure 2: RDU VIP Setup
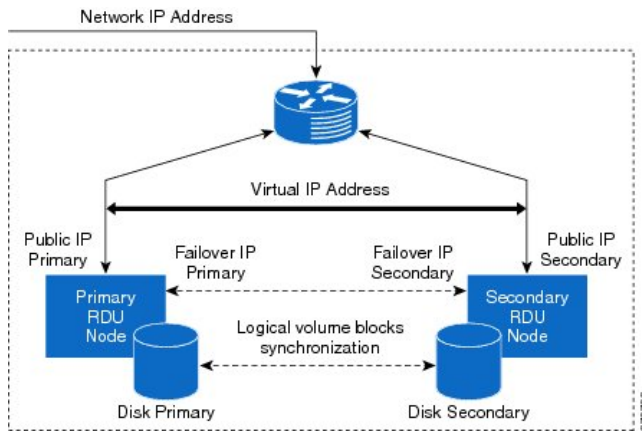
> **Note** No Intelligent Platform Management Interface(IPMI) controller is configured through RDU redundancy setup. (STONITH is not configured)

VIP and Interface Redundancy

The purpose of the VIP and interface redundancy feature is to provide IP addresses that can float between the nodes and provide redundancy between the two nodes. These interfaces perform the following:

- Provide network redundancy for interface addresses between both the nodes. The interface address has to be in the range of a subnet common to each RDU node.

- For Geo redundancy the VIP can be under any subnet.

- Provide redundancy for VIP addresses between nodes. VIP redundancy can be active or passive, where only one node services requests for the VIP, or shared, where multiple nodes service VIP requests.

- Both the IP addresses and MAC addresses of a redundant interface or VIP are shared. In other words, the MAC address does not change when the backup takes over.

- RDU redundancy setup can be reached via VIP and the VIP can be mapped to a domain name.

- No changes are required at the client layer to communicate to the RDU redundancy setup. During any failover and failback operation, clients may experience a short outage of RDU (maximum of 1 minute). This is the startup time required for RDU to come online on any node.

For more information on network redundancy and how to configure VIP, see Cisco Prime Cable Provisioning 5.3 Quick Start Guide.

bprAgent and RDU Process

RDU process is managed by bprAgent, the watchdog process that controls the state of the various Prime Cable Provisioning processes. Admin UI and SNMPAgent are two integral and important processes running along with the RDU. RDU HA provides redundancy for these processes. Failover events for RDU also migrates these processes along with it to the active node.

RDU HA compliance makes the following three set of resources redundant along with the bprAgent watchdog process:

1 bprAgent

   a RDU

   b Admin UI (tomcat server)

   c SNMPAgent

2 VIP

3 File systems. These are RDU redundancy specific file systems mounted on the synchronized logical volume.

   a /bprData

   b /bprHome

   c /bprLog

File System Replication

For Prime Cable Provisioning HA compliance, file system replication works on top of file blocks, which are LVM's (Logical Volume Manager) logical volumes (/bprData, /bprHome and /bprLog mounted over respective logical volumes). It mirrors each data block that is written to disk to the peer node.

In Prime Cable Provisioning, asynchronous mirroring is implemented. This means that the entity that issued the write requests is informed about completion as soon as the data is written to the local disk. Asynchronous mirroring is necessary to build mirrors over long distances, i.e., the interconnecting network's round trip time is higher than the write latency you can tolerate for your application.

> ✎
>
> **Note**   The network latency between primary and secondary node must not be more than 100 milliseconds.

Figure 3: Database Mirroring
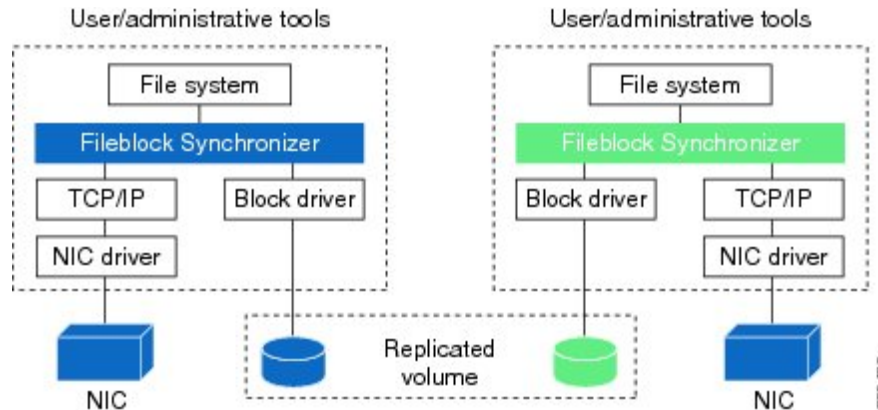
A consequence of mirroring data on block device level is that you can access your data, using a file system, only on the active node. This is not a shortcoming of Synchronizer but is caused by the nature of most file systems (ext3, XFS, JFS, ext4 ...). These file systems are designed for one computer accessing one disk, so they cannot cope with two computers accessing one (virtually) shared disk.

Prime Cable Provisioning uses the Logical Volume based synchronization and creates three logical volumes, which can be synced over the secondary. You could choose to have even one or two logical volumes.

- Logical volumes on both the nodes should be of same name and capacity. Following are the recommended configurations:

  ○ lv_bprHome(mounted on /bprHome , capacity 5 GB)

  ○ lv_bprData(mounted on /bprData , capacity 75 GB)

  ○ lv_bprLog(mounted on /bprLog , capacity 5 GB)

- Logical volumes must be pre-created with ext4 file system on them. The block size is set to default.

File System Synchronizer

During installation or migration, there is a huge change in data and for synchronization between nodes to begin, you must wait until disks on both sides are not in UpToDate state. The possible disk states are listed in Disk States, on page 12.

The File system available on both the nodes is the DRBD file system. For more information on DRBD resources, see http://drbd.linbit.com/. If you find the synchronization is stopped, use the following command on both the nodes a few times to start the synchronization

```
#fs_ha_adjust.sh all
```

Heartbeat Configurations

The heartbeat manager manages the heartbeat between the active and passive nodes. There are two physical interfaces connected on both nodes. Heartbeats are configured on both the network links on both nodes.

- Public network interface link—used to access the node by the external clients. For example, API client and PWS client

- Private network interface link—used for dual ring configuration

- Failover network interface link—failover network interface link/cross over cable used for synchronization data between the nodes

HA Cluster Management

CRM or cluster resource manager manages HA clustering when:

- RDU process is inactive. If RDU is not able to start due to some reason, after a preconfigured timeout RDU process should failover to secondary node.

- RDU becomes unresponsive. RDU process can become temporarily unresponsive for reasons like; the server is overloaded or the underlying database is corrupted. In situations like this the cluster resource manager is configured to do the following:

  ◦ If RDU process is able to respond before the timeout, do not failover.

  ◦ If RDU process remains unresponsive for longer than timeout, declare the existing primary RDU process unresponsive, and failover to secondary after the restart counts exceeds the threshold value (default 3 minute). Once the primary become responsive again failback to primary after a manual clean up of resources on primary node.

Changing Configuration for RDU Cluster Maintenance

Post installation, if you want to change or add any configuration properties to RDU, you can do so by:

Step 1   Use the VIP to SSH into the RDU server.

Step 2   Provide a valid root username and password to log in. The user must have the administrative privilege.

Step 3   Stop the RDU resource from CRM, using the command:

```
manage_ha_resource.sh stop res_bprAgent1
```

Step 4   Make the property configuration changes and then start the RDU using the command:

```
manage_ha_resource.sh start res_bprAgent1
```

Step 5   Verify that RDU is started:

```
/etc/init.d/bprAgent status
```

⚠️

Attention   Do not stop bprAgent using the /etc/init.d/bprAgent stop command. Doing so misguides the cluster manager state machine.

Following are some important configurations, whose details can be viewed using the command, monitor_ha_cluster.sh.

- Resources type

- VIP(res_IPaddr2_1) for local redundancy

- VIP(res_VIPArip) for Geo redundancy

- Master/slave resource for file system (res_drbd_1, res_drbd_2 and res_drbd_3)

- FileSystem resources for mounting the drbd on filesystem (res_Filesystem_1, res_Filesystem_2 and res_Filesystem_3)

- bprAgent resource (res_bprAgent_1)

- Health of each resource

- Failure timeout values

- Failure threshold

- Co-location of the resources (dependencies)

- Location of all the resources

Example

```
# BPR_HOME/agent/HA/bin/monitor_ha_cluster.sh
============
Stack: openais
Current DC: bac-c6-vm21 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
11 Resources configured.
============

Node bac-c6-vm21: online
        res_drbd_1:0    (ocf::linbit:drbd) Master
        res_Filesystem_1        (ocf::heartbeat:Filesystem) Started
        res_drbd_2:0    (ocf::linbit:drbd) Master
        res_IPaddr2_1   (ocf::heartbeat:IPaddr2) Started
        res_Filesystem_3        (ocf::heartbeat:Filesystem) Started
        res_Filesystem_2        (ocf::heartbeat:Filesystem) Started
        res_drbd_3:0    (ocf::linbit:drbd) Master
        res_bprAgent_1  (lsb:bprAgent) Started
Node bac-c6-vm22: standby

Inactive resources:

 Master/Slave Set: ms_drbd_1 [res_drbd_1]
     Masters: [ bac-c6-vm21 ]
     Stopped: [ res_drbd_1:1 ]
 Master/Slave Set: ms_drbd_2 [res_drbd_2]
     Masters: [ bac-c6-vm21 ]
     Stopped: [ res_drbd_2:1 ]
 Master/Slave Set: ms_drbd_3 [res_drbd_3]
     Masters: [ bac-c6-vm21 ]
     Stopped: [ res_drbd_3:1 ]

Migration summary:
* Node bac-c6-vm22:
* Node bac-c6-vm21:
```

```
    res_bprAgent_1: migration-threshold=3 fail-count=1 last-failure='Wed Sep 25 03:22:46
2013'

Failed actions:
    res_bprAgent_1_monitor_20000 (node=bac-c6-vm21, call=24, rc=7, status=complete): not
running
```

Cluster Timeout Configuration

Cluster components have a exponential back-off timeout for each of the resources. Here the file system resource is the most fundamental resource, which has to be functional before the bprAgent. Once the bprAgent resource is up and ready to serve only then you can configure the VIP and make it available for the rest of the service infrastructure. Timeout of the individual resources are as follows:

Table 1: Resources Timeout Value

Resources	Timeout value
All resources	Default failure threshold configured to three. After three attempts, resource with all dependencies will failover to peer node. The failed node is troubleshooted and cleaned up using utility scripts; the resource still continues to be alive on the secondary node until the failover timeout expires (default 30 minutes). On the expiry of failover timeout all resources failback to the primary node (preferred location). This happens only if auto-failback is enabled.
VIP address resource	• start interval="0" timeout="30 sec" • stop interval="0" timeout="20 sec" • monitor interval="10" timeout="20 Sec" start-delay="0"
bprAgent resource	• start interval="0" timeout="180 sec" • stop interval="0" timeout="180 sec" • monitor interval="30 sec" timeout="120 sec " start-delay="15 sec"
master-slave and file system resource for all three file system resources (/bprHome, /bprData and /bprLog)	• start interval="0" timeout="30 min" • promote interval="0" timeout="30 min" • demote interval="0" timeout="30 min" • stop interval="0" timeout="30 min" • monitor interval="10" timeout="20 Sec" start-delay="0" • notify interval="0" timeout="90 Sec"

For more information on these resources, see Cisco Prime Cable Provisioning 5.3 Quick Start Guide.

Disk States

Disk synchronization means synchronizing logical volumes between primary and secondary RDU disks (at file block level). Using Logical volume manager, you can create the logical volume group with three logical volume blocks on each disk, and these block devices would be mounted on home, data, and database log directories. These logical volume blocks are further configured on the File system driver to enable synchronization between RDU nodes.

You can verify the synchronization status using the script monitor_fs_sync_status.sh. This script can be run on either primary or secondary RDU server. The filesystem synchronizer resources displays the disk status in the format *<local disk status>/<remote disk status>* which means the local disk state is displayed first followed by the remote disk state. The disk state helps you to determine the synchronization status of primary and secondary RDU disks.

In RDU HA cluster, the disk state represents the filesystem state, and may be of the following type:

- UpToDate/UpToDate—Indicates that the disks are synchronized and file system resources are functioning normal.

- UpToDate/outdated—Indicates that the local server disk is updated but the remote disk is not synchronized properly. In this case, you must verify whether the File system driver is functioning normal and reinitiate the synchronization process.

- Attaching/Attaching—Indicates that the synchronization is in progress. the primary server is trying to achieve the network connectivity with the secondary server using the failover IP. You must wait until the synchronization completes before running any tasks on the RDU.

- Negotiating/Negotiating—Indicates that the network connectivity between primary RDU server is established, and the data synchronization is in progress. You must wait until the synchronization completes before running any tasks on the RDU.

- Failed /<any state>—Indicates that the synchronization is failed. Verify whether the logical volume blocks are configured on the File system resource driver. If not, set up the file system driver and reinitiate the synchronization process.

- <any state>/Inconsistent—Indicates that either a new resource is added on the local disk or the synchronization is in progress. Verify if any new resource is added before completion of the initial full synchronization. If yes, reinitiate the synchronization process.

- UpToDate/Unknown—Indicates that the network connectivity is not available between RDU nodes. Verify whether the failover IPs are correct and configured appropriately.

- Consistent/Consistent—Indicates that the data is consistent in both the disks based on the initial synchronization. To get the current disk status, initiate the synchronization process to verify whether

the data is up to date in both the disks. You may also receive this disk state if the synchronization is in progress.

Figure 4: Disk Space
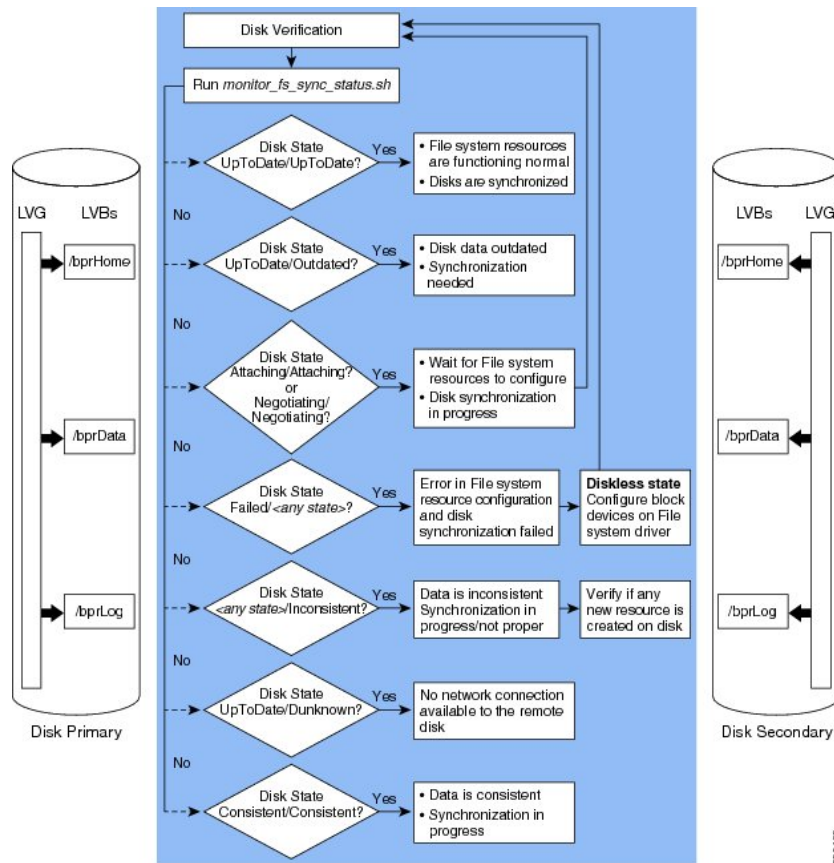
If the filesystems of primary and secondary RDU nodes are properly synchronized, you will observe the disk status as **UpToDate/UpToDate**.

Dual Ring Support on the Cluster

Dual Ring is configured to provide service redundancy for RDU HA setup. The RDU HA setup consists of the following network interfaces:

- Public interface–Network interface that provide public access to the RDU nodes through VIP

- Private interface–Network interface that provides the failover connectivity between RDU nodes, and supports file system synchronization.

The dual ring support in RDU HA setup uses the Redundant Ring Protocol (RRP). For information on RRP, see the RedHat Customer Portal.

The dual ring configuration involves deploying the following two network rings on these interfaces:

- A network ring on public interface that comprises of the pubic IP addresses of both the RDU nodes.

• A network ring on private interface that comprises of the failover IP addresses of both the RDU nodes.

The following figure describes the dual ring implementation in RDU HA setup:

Figure 5: Dual Ring Support
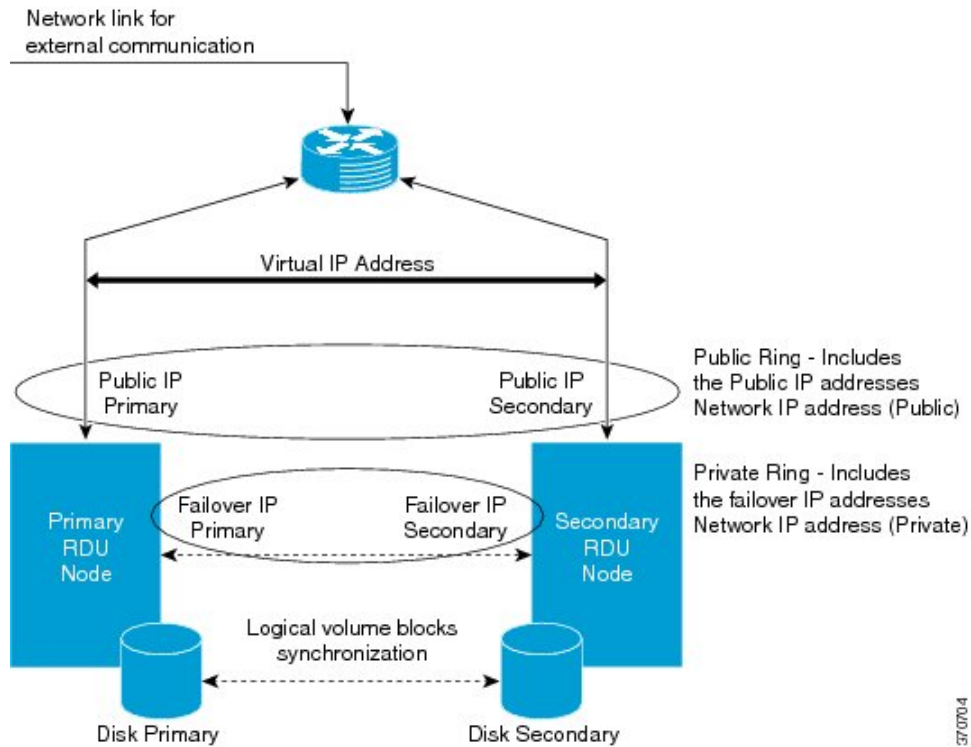
Earlier in RDU HA setup, only one network ring or totem ring was configured over public interface. Hence, during network interface failure, there was no communication channel available between the RDU HA cluster and the impacted RDU node. This lead to the instances of the required utilities getting terminated on the impacted RDU node. On revival of the network interface, it was required to restart these instances or the impacted node itself to reestablish the RDU HA cluster.

With the availability of dual ring support, if the network interface fails, the RDU HA cluster can still communicate with the impacted RDU node using the network ring on private interface. The communication between the RDU HA cluster and impacted RDU node is required to keep the instances of the required utilities alive on the impacted RDU node. For example, if the network ring on the public interface fails or the network connectivity to the primary RDU node is lost, the failover occurs and the secondary RDU node becomes active. In this situation, only if the Dual Ring support is available, the RDU HA cluster can communicate with the primary RDU node keeping the required utilities alive on the primary RDU node. Once the network connectivity is reestablished for the primary RDU node, the failback (if configured) occurs, and the primary RDU node becomes active with the required utilities running normal. Hence, the dual ring configuration helps you to avoid the manual restarting of the cluster processes on the primary RDU node.

Both these network rings are enabled in the system to maintain the RDU cluster synchronization. For dual ring support, you need to set the network IP addresses for both the rings. The network IP addresses that you configure must be applicable to both primary and secondary RDU nodes.

Split Brain Recovery

Split brain is a situation where the file systems of both the RDU nodes claim to be either in the same state; active or passive, or one exists in active with the data discrepancy between RDU nodes. Split brain occurs due to the following reasons:

- Loss of network connectivity on public or private interface between RDU nodes.

- High network latency between the primary and secondary RDU nodes during failover event.

- Cluster decision clash and data discrepancy in the file system synchronizer. This situation usually occurs when the network interface to the primary RDU node fails, and the primary RDU node loses the connectivity with the RDU HA cluster.

The following figure describes the split brain situation that occurs due to the cluster decision clash and data discrepancy in file system synchronizer:

Figure 6: Split Brain Occurrence
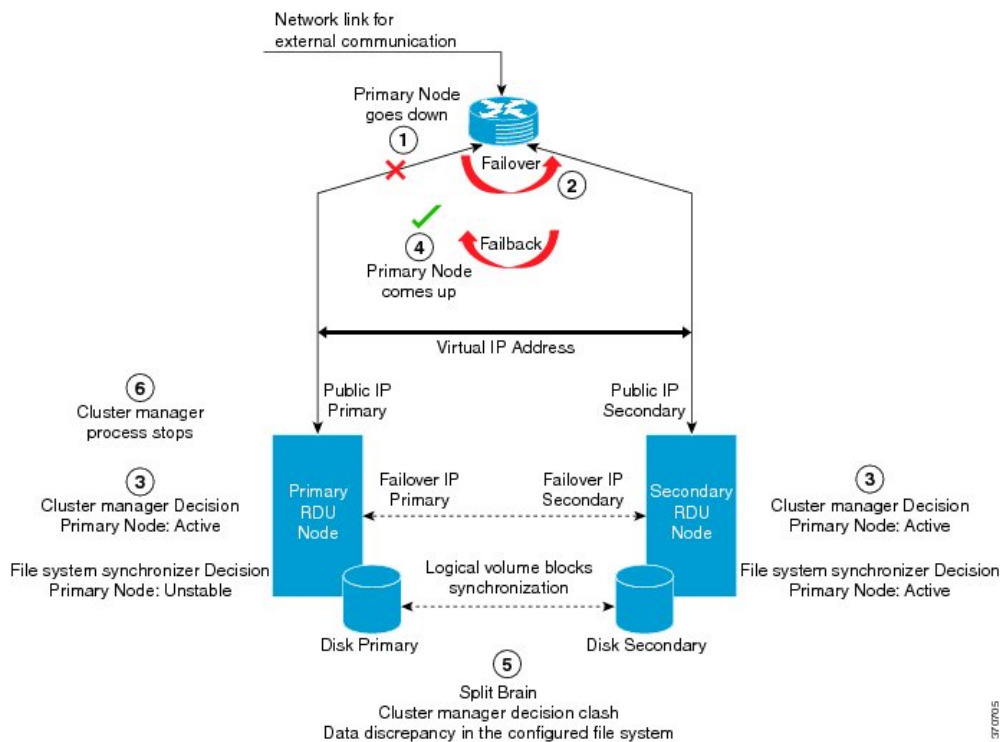
The set of instances that lead to the cluster decision clash and data discrepancy in file system synchronizer are:

1 The network connectivity to the primary RDU node fails.

2 The failover occurs and the secondary RDU node becomes active.

3 Since the communication between the primary RDU node and RDU HA cluster is lost, the cluster process and file system synchronizer resources claims the following status:

- Cluster on primary node: Claims primary node as Active

- Cluster on secondary node: Claims secondary node as Active

- File system synchronizer resources on primary node: Claims primary node as Unstable

- File system synchronizer resources on secondary node: Claims secondary node as Active

4 The primary RDU node comes up and the automatic failback (if configured) occurs.

| **Note** | If automatic failback is configured, you may also come across a situation where one file system synchronizer resource; for example, /bprLog claims to be active in secondary RDU node and other resources; for example, /bprHome and /bprData claims to be active in primary RDU node. |

5 Both the RDU HA cluster and file system synchronizer runs into split brain situation due to the cluster decision clash and the data discrepancy in the file system synchronizer.

6 The cluster process stops on primary RDU node.

You can avoid this split brain situation using the dual ring setup. For information on dual ring configuration, see Dual Ring Support on the Cluster, on page 13.

When the split brain occurs, one of the following split brain situations might arise:

- 0 primary–Both the RDU nodes claim as secondary (passive).

- 1 primary–Either of the RDU nodes claim as primary (Active), but there is a data discrepancy in the file systems between the RDU nodes.

- 2 primary– Both the RDU nodes claim as primary (active).

To recover from the split brain situation, you can define various automated policies. These policies help the system to determine the Split brain victim and split brain survivor, and accordingly resolve the split brain situation.

The following table describes the policies that are applicable for each split brain situation:

Table 2: Split Brain Situation - Applicable Policies

Split Brain Situation	Applicable Policies
0 primary	• disconnect–No automatic recovery. Invoke the split-brain handler script to disconnect the connection • discard- younger -primary–Discard changes on the RDU node that claimed the active role last before network failure. • discard-older-primary–Discard the older active RDU node, and retain the changes on the younger active RDU node. • discard-least-changes–Discard changes on the RDU node that has undergone fewer changes. • discard-zero-changes–If any node is found with zero changes, apply the changes that occurred on the other node on this non-impacted node.
1 primary	• disconnect–No automatic recovery. Invoke the split-brain handler script to disconnect the connection • Consensus–Select and apply the policy defined for 0 primary condition. If recovery fails, apply the disconnect policy. • call-pri-lost-after-sb–Select and apply the policy defined for 0 primary condition. If the split brain victim is found, invoke the pri-lost-after-sb handler script on it else apply the disconnect policy • discard-secondary–Discard changes on the RDU node that is not active.
2 primary	• disconnect–No automatic recovery. Invoke the split-brain handler script to disconnect the connection • violently-as0p–Select and apply the policy defined for 0 primary condition

In RDU HA setup, the following policies are configured for each split brain situation:

- For 0 primary–discard-older-primary
- For 1 primary–discard-secondary
- For 2 primary–violently-as0p

These policies help in retaining the most recent changes on the file system synchronizer. Also, these policies ensure that no data is lost during the split brain recovery process.

RDU HA Notifications

In Prime Cable Provisioning, you can configure the e-mail addresses of the recipients to receive RDU HA cluster notifications. The e-mail addresses of multiple recipients are configured using comma separated list. You can also configure a valid mailing-list to trigger the RDU HA e-mail notifications to a dedicated group of recipients.

The e-mail notifications are triggered for the following events:

- Primary or Secondary RDU node changes its state during the failover or failback occurrence.

- CRM resources become unresponsive or changes its state

- Split brain occurrence

This helps you to manage the system performance, and take corrective actions whenever required. For information on how to configure e-mail notifications in RDU HA setup, see Cisco Prime Cable Provisioning 5.3 Quick Start Guide.

Primary-Only and HA Ready

While installing RDU HA, you can also select the Primary-Only mode of installation. This makes the installation HA ready, so that you can later add secondary node and configure the HA cluster.

Recovery of Nodes in HA

If any of the RDU nodes in the HA cluster get corrupted, you can recover the impacted RDU node using the Recovery mode. The Recovery mode facilitates you to synchronize the impacted RDU node with the active RDU node, and restore the corrupted filesystem data.

RDU Geo Redundancy

RDU Geo Redundancy is an enhanced feature of RDU HA supported on RHEL 6.5 or CentOs 6.5 (both 64bit), wherein the RDU primary and secondary node can be in different geographical location or both the nodes can be in different subnet.

- In Geo redundancy mode the VIP can be in any subnet it is not necessary to have in the subnet range common to both nodes.

- In Geo redundancy mode the CIDR value of VIP should be 32

- The VIP will be advertised as a RIP advertisement from the active server, so on the ingress router of both the nodes route injection need to be done.

• In Geo redundancy mode, the VIP will be monitored using the resource agent (res_VIPArip).

Figure 7: RDU Geo Redundancy
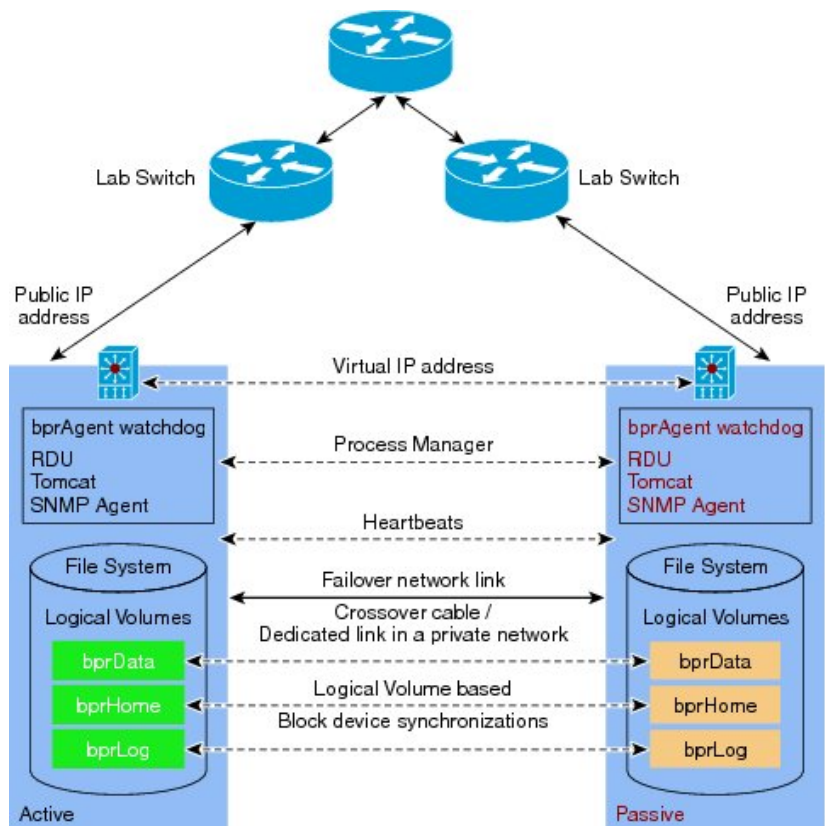
The solid line indicates physical network connectivity between the nodes. The dotted lines indicate logical synchronization between the two nodes.

RBAC Management

For better user management and security, Prime Cable Provisioning introduces Role Based Access Control (RBAC) that provides an approach to restrict access to system functions and resources to authorized users. Roles are composed of fine grain privileges. A privilege is a base unit of enforcement. A role groups a set of privileges into a logical job function that enables the customization of authorization policies.

Prime Cable Provisioning provides default out of the box (OOTB) roles, privileges, users, user groups, and domains that you can leverage from. Apart from these default configurations, you can also define your own setup to meet your organization requirements. The default OOTB configurations cannot be edited or deleted.

Authorization enforcement requires knowing the identity of the users and their granted privileges for any operation or resource to be protected. This information is used while performing access enforcement checks. There are four levels of checks.

• URL access check - Enforcement done by web facing components such as the Admin UI or web services.

- Operation/Method level check - Enforcement done by the components protecting access to operations. This type of access check is primarily performed in the RDU and DPE CLI. It is meant to ensure that the user has the correct privileges to invoke operations.

- Instance level check - Enforcement to ensure that the user has access to a specific object. This enforcement is performed in the RDU and leverage database capabilities.

- Property level check - Enforcement to ensure that the user has write access to a specific property. This enforcement is performed in the RDU.

For more details on RBAC configuration, see Configuring RBAC Using Admin UI.

Following topics are explained in this section:

- Authentication
- Authorization
- Role Evaluation
- Operation Level Access Control
- Instance Level Access Control
- Sample RBAC User Role Domain Hierarchy
- Property Filtering and Property Enforcement

Authentication

Authentication is the process of establishing the identity of a user. This process is achieved through the use of username/password credentials against the local RDU database or an external Radius server. Credentials are first checked in the Radius server and if not found, it is sent to the local server for validation. Radius authentication is possible only if it is enabled and configured in RDU Defaults. After validating the user's credentials, either an exception is shown in case of an authentication failure or in case of a valid user, the user is granted the privileges and domains based on the Roles, User-Groups, and Domains associated with the user.

Authorization

Prime Cable Provisioning users are authorized based on the various roles that are assigned to them directly or indirectly via user-groups. These roles consist of finer grain privileges. Following are the major authorization entities of Prime Cable Provisioning.

User

A user represents an identity that can either be a person or a system actor that is granted access to Prime Cable Provisioning. Depending on the role to be performed, a user can either have zero or more roles.

User Group

A user group is a collection of users. Like a user, a user group can also be assigned to zero or more roles. A user who is a member of a user group will inherit all the roles that the user group is assigned to. Those roles are constrained to only be valid on the resources that are also members of the group. A user can be a member of zero or more groups. The set of privileges the user gains is the aggregate of all those from the role.

Privilege

A privilege represents an authority granted for an operation that can be performed. It is the unit of enforcement. Privileges are grouped in roles, which are assigned to users. Privileges can have create, read, update, or delete actions.

Role

A role is a job function that defines a set of capabilities a user or user group can perform. A role binds privileges, users/user groups, and domains together. Prime Cable Provisioning comes with a set of default out-of-the-box roles and it also has the ability to create custom roles. Any set of privileges can be assigned to a custom role.

Domain

A domain represents a collection of objects (e.g. Device, COS, DHCP Criteria, Files, ProvGroup, etc) grouped for the purpose of instance level access control. The following are characteristics of domains:

- They are a set of instances. Domains can partition the overall system. A domain can have various object types. Authenticated users with the appropriate access privileges should be able to view the instances that exist in their domains.

- Domains are hierarchical. A user who has access to a parent domain can access all of the child domains, grandchild domains, and so on, of that parent.

- Domains have unique names across the system.

- A system defined (built-in) RootDomain can be used to give access to all objects for a particular role.

- Resources can be assigned to any domain. Resources can also be moved between domains. In the absence of domain assignment, the resources are assigned to the RootDomain.

Role Evaluation

A user can be granted privileges by directly assigning roles to the user or indirectly through user group membership. The total set of granted capabilities a user has is the union of all privileges derived from all roles directly or indirectly assigned to a user. The order of role evaluation is: user group, then user.

Operation Level Access Control

Every task that you carry out in Prime Cable Provisioning goes through an access control check. During the access control check, the task is validated against the privileges that you are assigned to and only if you have the right privilege, the task gets executed. In case you do not have the privilege, an insufficient privilege message is shown.

Instance Level Access Control

While operational level access control defines what actions a user can perform, instance level access control determines whether or not those actions can be performed on a specific instance. This is additional access control enforcement beyond checking of operation access. Operational access control must first be granted before instance level can be attempted. Prime Cable Provisioning supports a mode where operation level enforcement is enabled, but instance level access control is disabled.

In Prime Cable Provisioning, domains define the subset of instances that a user can access. Both the user and the resource must be members of the same domain. Otherwise, the user is not allowed to perform the operation. User's can be associated with zero or more domains. A resource (e.g. Device, COS, File, etc) can only be

associated with a single domain. When a resource is added to Prime Cable Provisioning, it is assigned to a domain. If no domain is assigned, the resource is automatically be added to the RootDomain.

Instance level access is controlled through RDU APIs associated with each resource.

Instance level access control can be enabled or disabled, using the *Instance Level Authorization* check box in the RDU Defaults page in the Admin UI. This checkbox appears only when the /adminui/enableDomainAdministration is set to true in the adminui.properties file.

Prime Cable Provisioning only supports instance level access control for the following resources: Device, COS, File, DPE, NR, Prov Group, and DHCP Criteria.

If instance level access control is disabled:

- When a new resource is added, the resource is internally and automatically assigned to the RootDomain by the respective API without requiring the user to assign the resource to a specific domain.

If instance level checking is enabled:

- When a new resource is being added, it is mandatory to provide a domain name to which the resource must be associated.

- The user must have access (i.e. membership) to the domain that a resource is being assigned to. This is enforced by APIs that assign or change the domain membership. The same API also enforces that a valid domain is being assigned to.

- In the Admin UI, if the user has access to a single domain, then that domain is selected by default while adding or modifying the resources.

- In the Admin UI, if the user has access to multiple domains, no default selection is made. The user needs to explicitly set the domain while adding a new resource.

- The user must have the PRIV_DOMAIN_READ privilege to read the configured domains. This privilege is added to all the out-of-the-box roles.

- The users will only be able to see the domains (includes sub-domains if any) that they are members of.

By default, newly added provisioning groups(PGs) are placed in the RootDomain. You can change a PG's default membership via the Admin UI or through the ChangeDomainProperties API. You can change a DPE's or CNR-EP's domain through the Admin UI or through APIs.

For a newly added DPE or CNR EP, it will take the domain membership of its PG. First primary PG in case of DPE.

While changing a PG's domain through the Admin UI, an option to apply the new domain to all the servers in the PG is provided. This can only be done through the Admin UI; there is no API support for this operation.

Note Configuration generation and regeneration does not support instance level access enforcement. The checks are only enforced at the administration operation and object level. For example, if a user is granted access to change a ClassOfService, the devices whose configuration may be regenerated are permitted as a result of that change. These devices do not undergo instance level access check enforcement during generation and regeneration.

Sample RBAC User Role Domain Hierarchy

Device Admin role contains the following privileges: COS read, DHCP criteria read, device create, read, update, delete.

File Admin role contains only File create, read, update, and delete privileges. Privileges are updated for deviceadmin and fileadmin roles.

Super Admin role can perform create, read, update, and delete on COS, DHCP criteria, device, and provisioning groups operations.

For example:

User W is assigned the Device Admin responsibilities for Domain 1. This allows user W to have:

- Read-only access for all COS in Domain 1.

- Read-only access for all DHCP Criteria in Domain 1.

- Create-Read-Update-Delete access on all Devices in Domain 1.

- Read access that permits the viewing of all details.

- Read access that permits the searching by MAC, DUID, FQDN, Owner ID, COS, DHCP Criteria etc.

- Update access that permits changing MAC, DUID, HOST NAME, Owner ID.

- The privilege to read a COS or DHCP Criteria plus being able to create-update a device. It also allows user W to assign or unassign COS and DHCP Criteria on those devices.

- If user W has read-update on device but no read access on COS, they will not be able to assign a COS, but they can still see the name of the assigned COS.

- Complete access to add, update, or delete any property on those devices in Domain 1.

- Create or update access implicitly enables user W to perform generate and regenerate operations. Having only read access will not be satisfactory.

- The PRIV_DEVICE_OPERATION privilege facilitates user W to invoke performOperation API requests.

If user X is assigned the Super Admin responsibilities for Domain 1, then user X can perform all operations that user W can along with the following operations:

- Create, read, update, and delete any COS, DHCP Criteria, Device in Domain 1.

- Access all provisioning groups associated with Domain 1 and see server details, access logs and perform DPE CLI operations.

If user Y is assigned Super Admin responsibilities for Domain Parent, then user Y can perform all operations that user X can and also have access to all of Domain Parent's child domains.

If user Z has similar capabilities as user X, but on Domain 3 then:

- User Z can create (add), read (view or export), update (replace or change properties), and delete files in either Domain Parent or Domain 3.

- Since user Z privileges include COS create, update on Domain 3, user Z can assign and unassign files to COSs that are associated with Domain 3.

DPE CLI Access Enforcement

Using the DPE CLI, you can view the status and configuration of the DPE as well as change properties. You must be authenticated to use the DPE CLI either through the local DPE, TACACS, or via Radius.

The local DPE CLI account has a specific username, admin and this requires only the local DPE CLI authentication. By default, the admin user enters into the disable mode upon authentication and then can enter into the enable mode without having to enter the password again. DPE CLI access is controlled by a set of DPE privileges. For details about DPE privileges, see Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide.

In Prime Cable Provisioning, a DPE audit log file is created to list the authentication details. This file is located at *BPR_DATA/dpe/logs*.

Property Filtering and Property Enforcement

To better manage the visibility of device level properties, Prime Cable Provisioning introduces a filtering mechanism in conjunction with write level access enforcement. This is an overly mechanism that is distinct from the fine grain access control capabilities discussed earlier. The Prime Cable Provisioning public APIs are extended to permit the filtering of device properties. It is the responsibility of the client to specify the filter. The Prime Cable Provisioning API call delivers only those properties that satisfy the filter. This mechanism is intended to both reduce the amount of data transmitted and to allow service based applications to enforce access control.

Prime Cable Provisioning provides write level access control for device properties. The list of properties that can be modified must be associated with a role. For a user to modify a given device property, it should be defined as part of the role to which the user is assigned to.

Note The write check for device properties is only at device level. A user can still add or change device properties at the higher property hierarchy like COS.

Configuration Regeneration Service (CRS)

When changes to the DHCP Criteria, Class of Service, group property or other such changes occur, device configurations become stale and require regeneration of the configuration. Prime Cable Provisioning provides a configuration regeneration service (CRS) that automatically regenerates configurations for all affected devices and sends the configurations to the DPEs. This eliminates the need to manually regenerate each configuration, and reduce the potential for introducing errors.

Device configurations are automatically regenerated whenever:

- The default Class of Service or DHCP Criteria for a technology default is changed.
- A Class of Service or DHCP Criteria property is changed.
- A group property is changed.
- A file related to a Class of Service or DHCP Criteria such as, CableLabsConfigTemplate, DocsisConfigTemplate, PacketCableConfigTemplate, and Script, is replaced.

Some configurations cannot be automatically regenerated because Prime Cable Provisioning cannot determine if the change impacts device configuration, such as:

• A technology default is changed, except for the default Class of Service and the default DHCP Criteria.

• The system defaults are changed.

• A file that is included within another DOCSIS template is changed.

• A Groovy class file and JAR file used by Groovy script is changed.

In such cases, manually regenerate configurations:

• From the Admin UI, on the Manage Devices page (see Regenerating Device Configurations).

• From the API *IPDevice.regenConfigs()*, for details, see the API Javadoc located at the docs directory of the build.

> **Note** Regardless of how configurations are regenerated, they are not propagated to the devices, unless the device reboots or device reset is triggered from RDU manually.

Prime Cable Provisioning provides greater control, error handling, and logging capabilities for the RDU CRS. You can now handle errors that are encountered during CRS request execution efficiently and monitor the operational status of CRS using the enhanced CRS logging and events. Using the Admin UI and RDU API you can manage CRS and CRS requests without restarting the RDU. For more information about configuring CRS using the Admin UI, see Configuring CRS.

Prime Cable Provisioning generates events when CRS is enabled, disabled, paused, and resumed. Events are also generated when a CRS request is created, deleted, replaced by an identical request, and when execution of a request is completed. These events enable you to monitor CRS as a system. Any external client that uses RDU API can be used to register and listen to all the CRS related events. The runEventMonitor.sh tool can also be used to listen to these events. For more information about this tool, see Using runEventMonitor.sh Tool. For more information on events, see the Cisco Prime Cable Provisioning 5.3 Integration Developers Guide.

Logging of CRS events are written into various log files to help debug and monitor CRS. The *rdu_crs.log* is introduced to capture all the CRS related activities in a compact manner. These CRS related logging messages are also displayed in the *rdu.log* when log level 6-Information is set. The *rdu.log* records logs from all the components, APIs, extensions, etc., this results in a large log file size and consequently, the data is rolled over to another file. As the *rdu.log* is very detailed, it is useful while debugging CRS failures. The *audit.log* captures all CRS related administrative details such as, CRS enable, disable, pause, resume, when a CRS request is deleted, event related information, and user associated with the event. For more information on logging see Monitoring Component Logs.

Handling Device Regeneration Failure

Prior to Prime Cable Provisioning 5.2.1, during execution of a CRS request if configuration regeneration for any device failed, CRS would retry the device configuration regeneration for that device endlessly. To avoid this issue, CRS now does not retry configuration regeneration for the failed device and proceeds to regenerate configurations for rest of the devices. Regeneration details about the failed devices and statistics are recorded in the *rdu_crs.log*. Details of the failed device regeneration are recorded after completing regeneration of every 1000 devices. The devices that are not present in database are ignored by the CRS and are not explicitly recorded in the *rdu_crs.log*. For more information on *rdu_crs.log*, see Regional Distribution Unit Logs.

Two new properties are introduced to efficiently manage the failed configuration regeneration. These properties can be set from the RDU Defaults of the Admin UI or using the `Configuration.changeRDUDefaults` API. Any changes made to these property values will take effect from the next CRS request.

- `failureThresholdPercentage`—Specifies the maximum acceptable percentage of failed devices. You could specify any value between 0.0 to 100.0%. By default this value is set to 0.0%.

- `pauseOnFailureThreshold`—Specifies whether CRS automatically pauses or not when percentage of failed devices exceeds `failureThresholdPercentage`. The acceptable values are *true* or *false*. By default `pauseOnFailureThreshold` is set to false.

   - When `pauseOnFailureThreshold` is set to true and once the percentage of failed devices exceeds `failureThresholdPercentage`, CRS automatically pauses and a warning message is logged in *rdu _crs.log*.

   - When `pauseOnFailureThreshold` is set to false and once the percentage of failed devices exceeds `failureThresholdPercentage`, CRS continues to regenerate configurations for rest of the devices and only a warning message is logged in *rdu _crs.log*.

During execution of a CRS request, the `failureThresholdPercentage` and `pauseOnFailureThreshold` properties are calculated after completing configuration regeneration of every 1000 devices.

For example, 2000 devices are associated with a CRS request, `failureThresholdPercentage` is set to 5.0% and `pauseOnFailureThreshold` is set to *true*. If the total number of devices that have failed so far has exceeded 100, then CRS will automatically pause. In this case, you can take one of the following corrective actions:

- Fix the issue by replacing with an identical request and resuming CRS manually. The regeneration starts from the beginning.

- Set the `pauseOnFailureThreshold` to false and resume CRS manually. This ignores the failed devices and the details are logged in *rdu_crs.log*.

- Set the `failureThresholdPercentage` to a higher value.

You can use the information available in the *rdu.log* to troubleshoot potential problems.

Clearing User Sessions

Once an RDU local user logs in, a user session is created. A user can have multiple concurrent sessions. The privileges and accessible domains of a user are cached until the last active session of the user is either terminated or timed out. When the admin changes the privileges or accessible domains of an RDU user, the new changes would not take effect until all existing sessions of that user are either terminated or timed out. This is not applicable for Radius users. Prime Cable Provisioning provides a shell script tool, closeSession.sh to clear all the sessions of a given user. This tool is available in the RDU under the location BPR_HOME/rdu/bin.

For example,

To clear the session of user John, run the command:

```
#./closeSession.sh John
```

Service-Level Selection

The extension point for service-level selection determines the DHCP Criteria and the Class of Service that the RDU is to use while generating a configuration for a device. The RDU stores this information for each device in its database.

The DHCP Criteria and the Class of Service that the RDU uses to generate a configuration for a device is based on the type of access granted to the device. Device access is of three types:

- Default—For devices granted default access, Prime Cable Provisioning uses the default Class of Service and DHCP Criteria assigned for the device type.

- Promiscuous—For devices granted promiscuous access, Prime Cable Provisioning obtains the Class of Service and DHCP Criteria from the relay agent that the device is behind.

- Registered—For devices granted registered access, Prime Cable Provisioning uses the Class of Service and the DHCP Criteria registered for the device in the RDU database.

There should always be one default extension per device type.

You can enter service-level selection extension points for specific technologies using the default pages at **Configuration > Defaults** from the Admin UI. For additional information, see Computer Defaults. By default, these properties are populated with zero or with one of the built-in extensions.

⚠️
Caution   Do not modify these extensions unless you are installing your own custom extensions.

Although a device may have been registered as having to receive one set of DHCP Criteria and Class of Service, a second set may actually be selected. The configuration generation extension looks for the selected DHCP Criteria and Class of Service and uses them.

The service-level selection extension selects a second Class of Service and DHCP Criteria based on certain rules that you specify for a device. For example, you may specify that a device must boot in a particular provisioning group for the device to be assigned a specific Class of Service and DHCP Criteria.

The extension returns information on why a specific set of DHCP Criteria and Class of Service is selected to provision a device. You can view these reasons from the Admin UI on the View Device Details page.

The following table describes these reasons and the type of access granted in that case.

Table 3: Reasons for Device Access as Determined by Service-Level Selection Extension

Reason Code	Description	Type of Device Access Granted		
		Default	**Promiscuous**	**Registered**
NOT_BEHIND_ REQUIRED_DEVICE	The device is not behind its required Cable Modem.	✓		
NOT_IN_REQUIRED_ PROV_GROUP	The device is not in its required provisioning group.	✓		
NOT_REGISTERED	The device is not registered.	✓		

Reason Code	Description	Type of Device Access Granted		
		Default	Promiscuous	Registered
PROMISCUOUS_ACCESS_ENABLED	Promiscuous access is enabled for the Cable Modem.		✓	
REGISTERED	The device is registered.			✓
RELAY_NOT_IN_REQUIRED_PROV_GROUP	The front device(Cable Modem) is not in the required provisioning group.	✓		
RELAY_NOT_REGISTERED	The front device(Cable Modem) is not registered.	✓		

Note Most of these reasons indicate violations of requirements for granting registered or promiscuous access, resulting in default access being granted.

Authentication Support

Authentication is the process of establishing the identity of a user to ensure that a user is who they claim to be. A user accessing the RDU can be authenticated locally by the RDU if they have an account created on the RDU. In addition, remote Radius authentication is supported. Prime Cable Provisioning supports the Authentication and Authorization features of AAA.

Local Authentication

This mode authenticates the user in the local RDU database and this mode is always enabled. For the admin user, only local authentication is used. For more details, see Adding a New User and RDU Defaults.

Remote Authentication

This mode authenticates the user in a remote server. Prime Cable Provisioning uses Radius and TACACS for remote authentication.

Radius Authentication

The RDU supports externalizing authentication to a remote Radius server. The user need not have an account in the RDU database. However, a reliable batch submitted by a Radius-only user cannot be guaranteed to execute across reboots or when the user logs out. This applies to those users that do not have their privileges defined in the RDU account but are only provided by remote Radius.

Radius authentication support can be configured using the Admin UI (Configuration > Defaults > RDU Defaults). To leverage Radius, a primary Radius server must be configured. This following properties must be provided: Primary Host, Primary Shared Secret, Primary Port (default=1812), Timeout (default=1000ms), Retries (default =1). If not specified, authentication will fail.

A secondary Radius server can also be configured. Only in the event the primary server is not available the secondary server is consulted to authenticate the user. The primary and secondary Radius servers support the same set of users.

Radius is a UDP-based protocol that supports centralized authentication, authorization, and accounting for network access. Radius authentication involves authenticating the users accessing the network services via the Radius server, using the Radius standard protocol defined in RFC 2865.

Radius authentication are of two modes and they are as follows:

- **Without Two-Factor:**

In this mode, username and password are required to log on to RDU which must be configured in Radius server.

Note   For the users authenticated via Radius, the password can be changed only in the Radius server.

- **Two-Factor:**

In this mode, username and passcode are required to log on to RDU. The username and assigning RSA SecureID Token to user must be configured in RSA Authentication Manager. The RSA SecureID generates the TokenCode which will be updated every 60 seconds in the RSA SecureID Token. The combination of TokenCode and the pin associated with the RSA SecureID Token will be used as the passcode of the user.

For example, if the PIN associated with the RSA SecureID token is 'user' and the Token Code generated from the RSA SecureID token is '12345', then the passcode is 'user12345'.

Note   Changing the combination order of the passcode from Token Code and PIN will result in authentication failure. The PIN for the RSA SecureID tokens must be assigned in RSA Authentication Manager through RSA Authentication Agents.

Creating or modifying a PIN for RSA SecureID token can be done via RSA Authentication Manager.

To enable Radius authentication, the authentication mode must be configured in the RDU Defaults page. For more details, see RDU Defaults.

Radius Integration

Prior to Prime Cable Provisioning, the property /rdu/auth/mode could take the value, LOCAL or Radius indicating which mode of authentication is enabled. With Prime Cable Provisioning, this is changed and the local mode is always enabled. /rdu/auth/mode is only used to enable or disable Radius authentication. If enabled, the external Radius servers are always used for authentication. If the servers fail to respond or reject the user, local authentication is attempted.

For Radius authenticated user, Prime Cable Provisioning supports granting privileges, sessions allowed, and domains to users through the Cisco-AVPair Radius VSA. The Radius Access-Accept can contain zero or more Cisco-AVPair VSA. The supported Cisco-AVPair format is:

Usergroup

Format: cp:groups=<group1>,…<groupN>

- Takes zero or more user group names.

- If no user group is assigned, the user is not granted any privileges.

- First it checks for mapping. If external to internal user group name mapping is found, privileges are given based on internal user group name. If there is no mapping, then the RDU database is searched for a user group with such a name. If that is also not there, then no privileges are assigned.

- The aggregate of all privileges is returned. This depends on the roles the user group possess.

- If no RDU database user group is found, no privileges are assigned.

Example `cp:groups=Administrators,Regional`

This specifies that the authenticated user is a member of two user groups: Administrators and Regional. The user is granted the roles assigned to these two groups. Also for a user to be granted privileges, the user group must be granted one or more roles.

Prime Cable Provisioning also supports mapping of external user group name to an internal user group name. See User Group Mapping for more details.

Sessions

Format: cp:sessions-allowed=<integer>

- The integer value must be zero or greater.

- If the value cannot be parsed, the user is given the RDU session default and an error is logged.

- If the Access-Accept does not specify the sessions-allowed, the user is given the RDU session default.

Example `cp:sessions-allowed=3`

If the Radius Access-Accept Cisco-AVPair VSA contains cp:sessions-allowed=3, the user is allowed three concurrent sessions.

Domain

Format: cp:domains=<domain1>,…<domainN>

- Domain names must explicitly match those in the RDU.

- Only those domains that match are assigned to the user.

- If no domains are specified, the user is not granted any domain memberships.

Example `cp:domains=north,south`

If the Radius Access-Accept Cisco-AVPair VSA contains cp:domains=north,south, the user is allowed membership into the north and south domains.

Backward Compatibility for Existing Users

Unlike in the earlier releases, there is no need to create duplicate users in both RDU and Radius in Prime Cable Provisioning. RDU user configuration overrides Radius user configuration for authorization. This is done to support backward compatibility of existing Radius users. After migrating from an earlier version to Prime Cable Provisioning, all existing Radius users are created as local users in RDU. So it is advised that you delete all the existing duplicate Radius users once the Radius users are configured with the appropriate Cisco AV Pairs. If there is a duplicate user (same name) present in both RDU and Radius, even though, the user would get authenticated by the Radius server, for authorization RDU configuration takes precedence.

For example:

If the user John is present in both Radius and RDU and in Radius, John is configured with COSAdmin privileges and in RDU, John is configured with DeviceAdmin privileges, on login using the Radius password, John is authenticated by Radius but the privileges set in Radius are ignored as the user configuration for John present in RDU takes precedence for authorization.

GSLB Support

Global Server Load Balancing (GSLB) directs DNS requests to the best-performing GSLB website in a distributed internet environment. In Prime Cable Provisioning, GSLB is used to implement failover, which enables the continuation of the RDU service after the failure of the primary RDU. When the primary RDU fails, all the client requests will be routed to the secondary RDU. In Prime Cable Provisioning, if the IP address of FQDN is changed and the primary RDU is down, FQDN is resolved to the new IP address and all the clients are routed to the secondary RDU.

Provisioning Web Service

The Provisioning Web Service (PWS) component of Prime Cable Provisioning provides a SOAP based web interface that supports provisioning operation. The provisioning services include functionalities such as: adding, retrieving, updating, and removing objects necessary to support the provisioning and configuration generation of CPEs. Here objects include devices, classes of service, DHCP criteria, groups, and files.

The web service is hosted on a tomcat container and it is recommended that you install it on a separate server and not on the RDU server.

> **Note**
> If you are installing both RDU and PWS on the same server, the installation configurations chosen for PWS take precedence over the Admin UI configurations. For example, if you have chosen secured mode of communication for Admin UI and non-secured mode for PWS, non-secured mode is chosen for both Admin UI and PWS.

For complete information about PWS, its APIs, capabilities and user cases see the Cisco Prime Cable Provisioning 5.3 Integration Developers Guide.

For details about PWS configuration, see Configuring Provisioning Web Services.

The web service manages these activities:

- Exposes a provisioning web service that provides functionality similar to the current API client.
- Supports stateless interactions.
- Supports both synchronous and asynchronous requests.
- Supports singular and plural operations.
- Supports both stop on failure and ignore on failure.
- Service interface supports request that can operate on multiple objects.
- Supports SOAP v1.1 and 1.2.
- Supports WSDL v1.1.
- Support WS-I Basic Profile v1.1.

• Support both HTTP and HTTPS transport.

Note PWS can communicate only with Prime Cable Provisioning RDU. It is not compatible with RDUs of earlier releases of Prime Cable Provisioning.

The following sections describe these PWS concepts:

Provisioning Web Services APIs

The following table lists the PWS APIs along with their descriptions. For more details about the APIs, see the Cisco Prime Cable Provisioning 5.3 Integration Developers Guide.

Table 4: PWS APIs

API name	Description
createSession	Authenticates a client and establishes a session between PWS client and the PWS.
closeSession	Releases the session between the user and the web service including closing the connection with Prime Cable Provisioning components.
addDevice	Submits a request for the addition of a new device.
addDevices	Submits a request to add multiple new devices. Using the execution options, each device can be wrapped in a separate transaction (batch) or a single transaction.
getDevice	Retrieves data associated with a specific device.
getDevices	Retrieves data associated with the specific devices.
getDevicesBehindDevice	Retrieves the list of devices downstream of a specific device. Either the device identifiers or the entire device object is returned.
getDevicesBehindDevices	Retrieves the list of devices downstream of the specified devices. Either the device IDs or the entire device object is returned.

API name	Description
updateDevice	Updates the properties of the specified device. **Note** While updating a device, if FQDN, host and domain details are provided, one of the details gets ignored. It is recommended not to provide all three details.
updateDevices	This operation applies the data contained in the specified device object to all devices specified in the lit of device IDs. This operation will apply the same update to all specified devices. Those properties excluded are: deviceIds, hostName, fqdn, embeddedDevices.
deleteDevice	Deletes the specified device.
deleteDevices	Deletes the specified devices.
unregisterDevice	Unregisters a device.
unregisterDevices	Unregisters multiple devices.
getDHCPLeaseInfo	Retrieves all known DHCP lease information about the specified IP address.
regenConfigs	Submits a request to regenerate configurations for the set of devices that match the specified search criteria.
rebootDevice	Reboots a device.
addDeviceType	Submits a request for the addition of a new device type.
getDeviceTypes	Retrieves all the device types available in the database.
updateDeviceTypes	Updates the specified device type.
deleteDeviceType	Deletes the specified device type.
deviceOperation	A generic operation that sends an opaque command and parameters to all specified devices.
addClassOfService	Submits a request for the addition of a new class of service.
getClassOfService	Retrieves data associated with the specified CoS.
updateClassOfService	Updates the properties of the specified CoS object.
deleteClassOfService	Deletes the specified CoS.
addDHCPCriteria	Submits a request for the addition of a new DHCPCriteria.
getDHCPCriteria	Retrieves the data associated with the specified DHCPCriteria.
updateDHCPCriteria	Updates the properties of the specified DHCPCriteria object.

API name	Description
deleteDHCPCriteria	Deletes the specified DHCPCriteria.
addFile	Submits a request for the addition of a new file.
getFile	Retrieves data associated with the specified file.
updateFile	Updates the properties or data of the specified file object.
deleteFile	Deletes the specified file.
addGroup	Adds a new group.
getGroup	Retrieves data associated with the specified group name.
updateGroup	Updates the properties of the specified group.
deleteGroup	Deletes the specified group.
pollOperationStatus	Queries for the status of RDU asynchronous requests specified by the transaction identifier.
Search	This is a generic operation. Retrieves devices, CoS, files, DHCPCriterion, or groups based on the search criterion defined in search object.

Asynchronous Service

Prime Cable Provisioning web service provides an asynchronous service to support asynchronous operations. Asynchronous service supports operations that are non-blocking for the client, i.e. client execution continues immediately after they submit asynchronous request, and the response is then processed later, likely by a different execution context or thread. The batch identifier can then be used to poll for the status of asynchronous or outstanding operations as well as the operations submitted as reliable batches.

Asynchronous service facilitates the following use cases:

- When response time is expected to be too long to wait for the response (For example, addDevices)

- When response time is not predictable

- When client needs non-blocking batch execution

To achieve effective asynchronous service calls, Prime Cable Provisioning ensures the following:

- Long persistence of the batch results for the asynchronous requests for later retrieval.

- Correlation between the requests and associated responses, at the RDU, PWS, and client side.

Session Management

A web services client must create a session for any communication with the PWS. To create a session with the PWS, the client must provide a valid context object in the request. A valid context object can either include information such as the RDU user's authentication information (username and password), the RDU details (host name and port number) that the request is targeted for or a unique session identifier. It could also have both the information. Upon authentication of these details, a session is created between the client and the PWS and the same session identifier is returned in the response.

PWS uses two types of sessions to communicate with the client:

- To interact with the PWS, a client must provide authentication information and identify the RDU the request is targeted for. Individual requests contain the user's username and password and the RDU information (host name and port details) included in a context object. This context object is sent to the PWS which it uses for authentication for every request that comes from the client.

- Alternatively, in case of multiple requests, a session between the client and PWS can be created and reused across all requests. Upon successful authentication, a session identifier is created and returned to the client. The client should use this identifier for all requests belonging to the same session. A session is per client per RDU and is bound to the capabilities of the roles and privileges assigned to the client. The context object identifies the client, contains client authentication token, and identifies the RDU that the client wants to interact with. All client requests must include the context.

  Upon completing the interaction, the client closes the session. The session also gets closed automatically after a configurable period of idle time. The default idle timeout is 15 minutes and can be changed using a CLI command. Sessions also get closed whenever the PWS is restarted.

Transactionality

The PWS operations can be classified into two categories:

- Single device operations - One operation for a single device. For single device operations, an operation is a single transaction in which all the changes are made or no change is made to the device.

- Multiple device operations - One operation for multiple devices. For multiple device operations, the transaction scope can be set to include all devices, or have each device change its individual transaction.

You can set the execution option, transactionPerItem, to true to retrieve the transaction data for each batch. The OperationStatus returned will contain the status of the entire operation as well as the individual status if multiple transactions are used.

Error Handling

Prime Cable Provisioning uses SOAP faults to relay information regarding issues encountered in validating, processing, or executing client requests. The primary exception raised by operations is the ProvServiceException. The code and message contained in the exception describes the cause of the fault. This exception is used to relay issue raised by the RDU. PWS provides an information level log.

Device Provisioning Engines

The Device Provisioning Engine (DPE) communicates with CPE to perform provisioning and management functions.

The RDU generates DHCP instructions and device configuration files, and distributes them to the relevant DPE servers. The DPE caches these DHCP instructions and device configuration files. The DHCP instructions are then used during interactions with the Network Registrar extensions, and configuration files are delivered to the device via the TFTP service.

Prime Cable Provisioning supports multiple DPEs. You can use multiple DPEs to ensure redundancy and scalability.

The DPE handles all configuration requests, including providing configuration files for devices. It is integrated with the Network Registrar DHCP server to control the assignment of IP addresses for each device. Multiple DPEs can communicate with a single DHCP server.

In the DPE, the configurations are compressed using Delta Compression technique of RFC 328 to reduce overall DPE cache size for better scalability.

The DPE manages these activities:

- Synchronizes with the RDU to retrieve the latest configurations for caching.
- Generates last-step device configuration (for instance, DOCSIS timestamps).
- Provides the DHCP server with instructions controlling the DHCP message exchange.
- Delivers configuration files via TFTP.
- ToD server
- Integrates with Network Registrar.
- Provisions voice-technology services.

Configure and manage the DPE from the CLI, which you can access locally or remotely via Telnet. For specific information on the CLI commands that a DPE supports, see the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide.

DPE Licensing

Licensing controls the number of DPEs (nodes) that you can use. If you attempt to install more DPEs than you are licensed to use, those new DPEs will not be able to register with the RDU, and will be rejected. Existing licensed DPEs remain online.

Note   For licensing purposes, a registered DPE is considered to be one node.

When you add a license or extend an evaluation license or when an evaluation license has expired, the changes take effect immediately.

When you delete a registered DPE from the RDU database, a license is freed. Because the DPEs automatically register with the RDU, you must take the DPE offline if the intention is to free up the license. Then, delete the DPE from the RDU database via the Admin UI or via the API.

Deleted DPEs are removed from all the provisioning groups that they belong to, and all Network Registrar extensions are notified that the DPE is no longer available. Consequently, when a previously deleted DPE is registered again, it is considered to be licensed again and remains so until it is deleted from the RDU again or its license expires.

DPEs that are not licensed through the RDU do not appear in the Admin UI. You can determine the license state only by examining the DPE and RDU log files (*BPR_DATA/dpe/logs/dpe.log* and *BPR_DATA/rdu/logs/rdu.log*).

Note The functions enabled via a specific license continue to operate even when the corresponding license is deleted from the system.

For detailed information on licensing, see Managing Licenses.

For important information related to DPEs, see:

- DPE CLI Authentication

- DPE-RDU Synchronization

- TFTP Server

- ToD Server

Also, familiarize yourself with the information described in Provisioning Concepts.

DPE CLI Authentication

There are two authentication modes used for DPE CLI:

- TACACS+ Authentication

- Radius Authentication

See the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide for details about how to configure TACACS+ and Radius authentication in DPE CLI.

TACACS+ Authentication

TACACS+ is a TCP-based protocol that supports centralized access for large numbers of network devices and user authentication for the DPE CLI.

Through TACACS+, a DPE CLI can support many users, with each username and password configured at the TACACS+ server. TACACS+ is used to implement the TACACS+ client/server protocol (ASCII login only).

TACACS+ Privilege Levels

The TACACS+ server uses the TACACS+ protocol to authenticate any user logging in to a DPE CLI. The TACACS+ client specifies a certain service level that is configured for the user.

The following table identifies the two service levels used to authorize DPE CLI user access.

Table 5: TACACS+ Service Levels

Mode	Description
Login	User-level commands at router> prompt.
Enable	Enable-level commands at router# prompt.

TACACS+ Client Settings

A number of properties that are configured from the DPE CLI are used for TCAACS+ authentication. For information on commands related to TACACS+, see the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide.

When TACACS+ is enabled, you must specify either the IP addresses of all TACACS+ servers or their fully qualified domain names (FQDNs) with non-default values.

You can also specify these settings using their default values, if applicable:

- The shared secret key for each TACACS+ server. Using this key, you can encrypt data between the DPE and the TACACS+ server. If you choose to omit the shared secret for any specific TACACS+ server, TACACS+ message encryption is not used.

- The TACACS+ server timeout. Using this value, you can specify the maximum length of time that the TACACS+ client waits for a TACACS+ server to reply to protocol requests.

- The TACACS+ server number of retries. Using this value, you can specify the number of times that the TACACS+ client attempts a valid protocol exchange with a TACACS+ server.

Radius Authentication

DPE CLI supports Radius authentication for authenticating the users logging on to DPE CLI. Radius authentication are of two modes and they are as follows:

Without Two-Factor:

In this mode, username and password are required to log on to DPE CLI.

Two-Factor:

In two-factor authentication mode, the user has to provide the username and, the passcode which is a combination of PIN and Token Code to log on to DPE CLI. The RSA SecureID generates the Token Code which will be updated every 60 seconds in the RSA SecureID device.

Radius Privilege Levels

The Radius server authenticates the user logging on to a DPE CLI. The Radius client settings specifies certain privilege levels that are configured for the user.

The following table describes the service levels used to authorize the DPE CLI user.

Table 6: Radius Service Levels

Mode	Description
Login	User-level commands at router> prompt.

Mode	Description
Enable	Enable-level commands at router# prompt.

DPE-RDU Synchronization

The DPE-RDU synchronization is a process of automatically updating the DPE cache to be consistent with the RDU. The DPE cache comprises the configuration cache, with configurations for devices, and the file cache, with files required for devices.

Under normal conditions, the RDU generates events containing configuration updates and sends them to all relevant DPEs to keep them up to date. Synchronization is needed if the DPE is missing some events due to connection loss. Such loss could be because of a network issue, the DPE server going down for administrative purposes, or a failure.

Synchronization also covers the special case when the RDU database is restored from backup. In this case, the DPE cache database must be returned to an older state to be consistent with the RDU.

The RDU and DPE synchronization process is automatic and requires no administrative intervention. Throughout the synchronization process, the DPE is still fully capable of performing provisioning and management operations on the CPE.

Synchronization Process

The DPE triggers the synchronization process every time it establishes a connection with the RDU.

When the DPE first starts up, it establishes the connection to the RDU and registers with the RDU to receive updates of configuration changes. The DPE and RDU then monitor the connection using heartbeat message exchanges. When the DPE determines that it has lost its connection to the RDU, it automatically attempts to re-establish it. It continues its attempts with a backoff-retry interval until it is successful.

The RDU also detects the lost connection and stops sending events to the DPE. Because the DPE may miss the update events from the RDU when the connection is down, the DPE performs synchronization every time it establishes a connection with the RDU.

During the process of synchronization, the DPE is in the following states:

1   Registering—During the process of establishing a connection and registering with the RDU, the DPE is in the *Registering* state.

2   Synchronizing—The DPE requests groups of configurations that it should have from the RDU. During this process, the DPE determines which configurations in its store are inconsistent (wrong revision number), which ones are missing, and which ones to delete, and, if necessary, updates the configurations in its cache. The DPE also synchronizes deliverable files in its cache for the TFTP server. To ensure that the RDU is not overloaded with configuration requests, the DPE posts only one batch at a time to the central server.

3   Ready— The DPE is up to date and fully synchronized with the RDU. This state is the typical state that the DPE is in.

The following table describes some other states that the DPE may be in from time to time.

Table 7: Related DPE States

State	Description
Initializing	Is starting up
Shutting Down	Is in the process of stopping
Down	Does not respond to queries from Network Registrar extension points
Ready Overloaded	Is similar to *Ready* except that there is a heavy load on the system on which the DPE is running

Note Regardless of the state that the DPE is in, it continues to service device configuration, TFTP, and ToD requests.

You can view the DPE state:

• From the Admin UI. See Monitoring DPE.

• From the DPE CLI using the **show dpe** command. See the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide for more information.

TFTP Server

The integrated TFTP server receives requests for files, including DOCSIS configuration files, from device and nondevice entities. This server then transmits the file to the requesting entity.

Enable the TFTP server in DPE to access the local file-system. The local files are stored in the *<BPR_DATA>/dpe/tftp* directory. All deliverable TFTP files are precached in the DPE; in other words, the DPE is always up to date with all the files in the system.

Note The TFTP service on the DPE features one instance of the service, which you can configure to suit your requirements.

By default, the TFTP server only looks in its cache for a TFTP read. However, if you run the **service tftp** *1..1* **allow-read-access** command from the DPE command line, the TFTP server looks in the local file system before looking in the cache. If the file exists in the local file system, it is read from there. If not, the TFTP server looks in the cache. If the file exists in the cache, the server uses it; otherwise, it returns an error.

When you can enable read access from the local file system, directory structure read requests are allowed only from the local file system.

Note Ensure that you give unique names to all TFTP files instead of differentiating the files by using upper or lowercase. The filename casing is important because the DPE, while looking for a file in its local directory or cache, converts all filenames to lowercase.

You can specify TFTP transfers using IPv4 or IPv6, using the **service tftp** *1..1* **ipv4** | **ipv6 enabled true** command from the DPE command line. You can also specify a block size for these transfers using the **service tftp** *1..1* **ipv4** | **ipv6 blocksize** command. The blocksize option specifies the number of data octets and allows the client and server to negotiate a block size more applicable to the network medium. When you enable blocksize, the TFTP service uses the requested block size for the transfer if it is within the specified lower and upper limits. For detailed information, see the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide for more information.

The TFTP service maintains statistics for the number of TFTP packets that are processed for TFTPv4 and TFTPv6. You can view these statistics from the Admin UI on the device details page. For more information, see Viewing Device Details.

ToD Server

The integrated time of day (ToD) server in Prime Cable Provisioning provides high-performance UDP implementation of RFC 868.

Note The ToD service on the DPE features one instance of the service, which you can configure to suit your requirements.

You can enable the ToD service to support IPv4 or IPv6, from the DPE command line, using the **service tod** *1..1* **enabled true** command. The ToD service is, by default, disabled on the DPE.

While configuring this protocol on the DPE, remember that the ToD service binds only to those interfaces that you have configured for provisioning. For detailed information on configuring the ToD service, see the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide for more information.

The ToD service maintains statistics for the number of ToD packets that are processed for ToDv4 and ToDv6. You can view these statistics from the Admin UI on the device details page. For more information, see Viewing Device Details.

Cisco Prime Network Registrar

Cisco Prime Network Registrar provides the DHCP and DNS functionality in Prime Cable Provisioning. The DHCP extension points on Network Registrar integrate Prime Cable Provisioning with Network Registrar. Using these extensions, Prime Cable Provisioning examines the content of DHCP requests to detect device type, manipulates the content according to its configuration, and delivers customized configurations for devices that it provisions.

For additional information on Cisco Prime Network Registrar, see the http://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/network_registrar/8-2/user/guide/CPNR_8_2_User_Guide.html; http://www.cisco.com/c/en/us/support/cloud-systems-management/prime-network-registrar/products-command-reference-list.html; and http://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/network_registrar/8-2/installation/guide/CPNR82InstallGuide.html.

DHCP

The DHCP server automates the process of configuring IP addresses on IP networks. The protocol performs many of the functions that a system administrator carries out when connecting a device to a network. DHCP automatically manages network-policy decisions and eliminates the need for manual configuration. This feature adds flexibility, mobility, and control to networked device configurations.

This Prime Cable Provisioning release supports DHCP for IPv6, also known as DHCPv6. DHCPv6 enables DHCP servers to deliver configuration parameters, via extensions, to IPv6 hosts. IPv6 hosts by default use stateless auto-configuration, which enables IPv6 hosts to configure their own addresses using a local IPv6 router. DHCPv6 represents the stateful auto-configuration option, a technique in which configuration information is provided to a host by a server.

DHCPv6 provides:

- Expanded addressing capabilities via IPv6 addresses

- Easy network management and administration using the stateful auto-configuration protocol

- Improved support for options and extensions

- DHCPv6 relay agent encapsulates all the incoming packets (from a client or another relay agent) into a Relay-Forward message. Hence, maintaining a clean separation between client and relay agent options which makes DHCPv6 more desirable compared to DHCPv4.

- Assignment of multiple addresses to one interface

DHCPv4 versus DHCPv6

Much like DHCPv4, DHCPv6 uses a client-server model. The DHCP server and the DHCP client converse with a series of messages to request, offer, and lease an IP address. Unlike DHCPv4, DHCPv6 uses a combination of unicast and multicast messages for the bulk of the conversation instead of broadcast messages.

Some other differences between DHCPv4 and DHCPv6 are:

- Unlike DHCPv4, IPv6 address allocation in DHCPv6 is handled using a message option.

- Message types, such as DHCP Discover and DHCP Offer supported by DHCPv4 are removed in DHCPv6. Instead, DHCPv6 servers are located by a client Solicit message followed by a server Advertise message.

- Unlike DHCPv4 clients, DHCPv6 clients can request multiple IPv6 addresses.

DHCPv4 failover allows pairs of DHCP servers to act in such a way that one can take over if the other stops functioning. The server pairs are known as the main and backup server. Under normal circumstances, the main server performs all DHCP functions. If the main server becomes unavailable, the backup server takes over. In this way, DHCP failover prevents loss of access to the DHCP service if the main server fails. Currently DHCPv6 failover is not supported.

DNS

The DNS server contains information on hosts throughout the network, such as IP address host names. DNS uses this information primarily to translate between IP addresses and domain names. The conversion of names such as www.cisco.com to IP addresses simplifies accessing Internet-based applications.

Cisco Prime Network Registrar provide separate DNS servers for authorization and caching. It also supports DNS64, DNSSEC and full IPv6.

DNSv4 versus DNSv6

Much like DNSv4, DNSv6 uses a client-server model. The DNS client requests the DNS server to resolve the DNS name, during conversation DNSv6 uses IPv6, whereas DNSv4 uses IPv4.

Some other differences between DNSv4 and DNSv6 are:

- DNSv6 uses new AAAA RR type for name to address information

- DNS updates are just like DNSv4, except multiple AAAAs under a single client name

- IN DNSv6 PTR RR are under ip6.arpa and in DNSv4 it is under in-addr.arpa

HA DNS allows pairs of DNS servers to act in such a way that one can take over if the other stops functioning. The server pairs are known as the main and backup server. Under normal circumstances, the main server performs all DNS functions. If the main server becomes unavailable, the backup server takes over. In this way, HA DNS prevents loss of Dynamic updates and query translations.

Lease Query

The lease query feature allows you to request current IP address information directly from the Network Registrar DHCP servers in a provisioning group. To find a device's IP address, the RDU sends DHCP lease query messages only to the DHCP servers in the device's provisioning group, which prevents querying all DHCP servers in the network. Among all the responses, the response from the server that last communicated with the devices is taken as the authoritative answer.

In earlier Prime Cable Provisioning versions, the lease query feature relied on the operating system to select the source interface and the source port for sending lease query requests. In Prime Cable Provisioning, you can configure the RDU to use a specific interface and source port.

> **Note** When you install all components in the same Linux server, Cisco Prime Network Registrar will not respond to the lease queries from RDU.

For detailed information on lease query support in this Prime Cable Provisioning release, see Lease Query.

Key Distribution Center

The Key Distribution Center (KDC) authenticates PacketCable MTAs and also grants service tickets to MTAs. As such, it must check the MTA certificate, and provide its own certificates so that the MTA can authenticate the KDC. It also communicates with the DPE (the provisioning server) to validate that the MTA is provisioned on the network.

The certificates used to authenticate the KDC are not shipped with Prime Cable Provisioning. You must obtain the required certificates from Cable Television Laboratories, Inc. (CableLabs), and the content of these certificates must match those that are installed in the MTA. For additional information, see Using PKCert.sh.

> **Caution** The KDC does not function if the certificates are not installed.

The KDC also requires a license to function. Obtain a KDC license from your Cisco representative and install it in the correct directory. For details on how to install the license, see KDC Licenses.

The KDC has several default properties that are populated during a Prime Cable Provisioning installation into the *BPR_HOME/kdc/<Operating System>/kdc.ini* properties file, for Solaris. For Linux, the path is */opt/CSCObac/kdc/linux/kdc.ini*. You can edit this file to change values as operational requirements dictate. For detailed information, see Default KDC Properties.

The KDC also supports the management of multiple realms. For details on configuring additional realms, see Configuring Additional Realms.

Note   KDC is only required for PacketCable Secure mode.

Process Watchdog

The Prime Cable Provisioning process watchdog is an administrative agent that monitors the runtime health of all Prime Cable Provisioning processes. This watchdog process ensures that if a process stops unexpectedly, it is automatically restarted. One instance of the Prime Cable Provisioning process watchdog runs on every system which runs Prime Cable Provisioning components.

You can use the Prime Cable Provisioning process watchdog as a command-line tool to start, stop, restart, and determine the status of any monitored processes.

See Prime Cable Provisioning Process Watchdog, for additional information on how to manage the monitored processes.

SNMP Agent

Prime Cable Provisioning provides basic SNMP v2-based monitoring of the RDU and DPE servers. The Prime Cable Provisioning SNMP agents support SNMP informs and traps, collectively called notifications.

You can configure the SNMP agent:

- On the RDU, using the SNMP configuration command-line tool (see Monitoring Servers Using SNMP) or via the API.

- On the DPE, using the **snmp-server** CLI commands. See the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide for more information.

Administrator User Interface

The Prime Cable Provisioning Admin UI is a web-based application for central management of the Prime Cable Provisioning system. You can use this system to:

- Configure global defaults

- Define custom properties

- Add, modify, and delete Class of Service

- Add, modify, and delete DHCP Criteria

- Manage CRS

- Add, modify, and delete devices

- Group devices

- View server status and server logs

- Manage users

- Manage user groups

- Manage roles

- Manage domain

See these chapters for specific instructions on how to use this interface:

- Chapter 4, "Accessing Admin UI," describes how to access and configure the Prime Cable Provisioning Admin UI.

- Chapter 11, "Configuring Prime Cable Provisioning Using Admin UI," provides instructions for performing administrative activities involving the monitoring of various Prime Cable Provisioning components.

- Chapter 18, "Provisioning Devices Using Admin UI," describes tasks that you perform to configure devices using the Admin UI.

- Chapter 12, "Configuring Groups Using Admin UI," describes tasks that you perform to configure groups using the Admin UI.

- Chapter 21, "Monitoring Servers Using Admin UI," describes tasks that you perform to check the status of Prime Cable Provisioning servers using the Admin UI.

- Chapter 13, "Configuring RBAC Using Admin UI," describes tasks that you perform to add users, groups and map them. It also describes role management and domain management.

Provisioning Concepts

This section describes those concepts that are key to provisioning and include:

- Provisioning Groups, on page 45

- Static versus Dynamic Provisioning, on page 46

- Provisioning Group Capabilities, on page 47

Provisioning Groups

A provisioning group is designed to be a logical (typically geographic) grouping of servers that usually consists of one or more DPEs and a failover pair of DHCP servers. Each DPE in a given provisioning group caches identical sets of configurations from the RDU, thus enabling redundancy and load balancing. As the number of devices grows, you can add additional provisioning groups to the deployment.

Note   The servers for a provisioning group are not required to reside at a regional location. They can just as easily be deployed in the central network operations center.

Provisioning groups enhance the scalability of the Prime Cable Provisioning deployment by making each provisioning group responsible for only a subset of devices. This partitioning of devices can be along regional groupings or any other policy that the service provider defines.

To scale a deployment, the service provider can:

- Upgrade existing DPE server hardware

- Add DPE servers to a provisioning group

- Add provisioning groups

To support redundancy and load sharing, each provisioning group can support any number of DPEs. As the requests come in from the DHCP servers, they are distributed between the DPEs in the provisioning group and an affinity is established between the devices and a specific DPE. This affinity is retained as long as the DPE state within the provisioning group remains stable.

Static versus Dynamic Provisioning

Prime Cable Provisioning provisions devices in the network using device configurations, which is provisioning data for a specific device based on its technology type. You can provision devices using Prime Cable Provisioning in two ways: static provisioning and dynamic provisioning.

During static provisioning, you enter static configuration files into the Prime Cable Provisioning system. These configuration files are then delivered via TFTP to the specific device. Prime Cable Provisioning treats static configuration files like any other binary file.

During dynamic provisioning, you use templates or scripts, which are text files containing DOCSIS, PacketCable, or CableHome options and values that, when used with a particular Class of Service, provide dynamic file generation. A dynamic configuration file provides more flexibility and security during the provisioning process.

The following table describes the impact of static and dynamic provisioning using the corresponding files.

Table 8: Static Provisioning versus Dynamic Provisioning

Static Provisioning Using Static Files	Dynamic Provisioning Using Template Files
Used when fewer service offerings are available	Used when many service offerings are available
Offers limited flexibility	Offers more flexibility, especially when devices require unique configurations
Is relatively less secure	Is more secure
Offers higher performance	Offers slower performance, because every time you update a template or a groovy script assigned to a device, configurations for all devices associated with that template are updated.

Static Provisioning Using Static Files	Dynamic Provisioning Using Template Files
Is simpler to use	Is more complex

Provisioning Group Capabilities

To provision a subset of devices in a deployment, provisioning groups must be capable of as well as enabled to provision those devices. For example, to provision a DOCSIS 3.0 modem in IPv4 mode, you must enable the IPv4 - DOCSIS 3.0 capability. Following is the list of prominently used capabilities:

* IPv4 - DOCSIS 1.0/1.1

* IPv4 - DOCSIS 2.0

* IPv4 - DOCSIS 3.0

* IPv4 - PacketCable

* IPv4 - CableHome

* IPv4 - ERouter 1.0

* IPv6 - DOCSIS 3.0

* IPv6 - PacketCable 2.0

* IPv6 - ERouter 1.0

In previous Prime Cable Provisioning releases, each DPE in a provisioning group registered what it was capable of supporting with the RDU at startup. After server registration, the provisioning group was automatically enabled to support the device types it was capable of supporting. In Prime Cable Provisioning, you must enable device support or capabilities manually.

- From the Admin UI, on the Provisioning Group Details page (see Monitoring Provisioning Groups).

- From the API, using the *ProvGroupCapabilitiesKeys* constants. For details, see the API Javadoc located at the docs directory of the build.

Component Based Log Files

Logging of events is performed at the RDU, DPE, KDC, and PWS, and in some unique situations, DPE events are additionally logged at the RDU to give them higher visibility. Log files are stored in their own log directories (*BPR_HOME/<component>/logs*) and can be examined by using any text processor. You can compress the files for easier e-mailing to the Cisco Technical Assistance Center or system integrators for troubleshooting and fault resolution. You can also access the RDU and the DPE logs from the Admin UI.

You can generate server configuration and other diagnostics information using the diagnostics tools in the *BPR_HOME/<component>/diagnostics/bin* directory. For details see Bundling Server State for Support.

For detailed information on log levels and structures, and how log files are numbered and rotated, see Log Levels and Structures.

Deployment of Prime Cable Provisioning

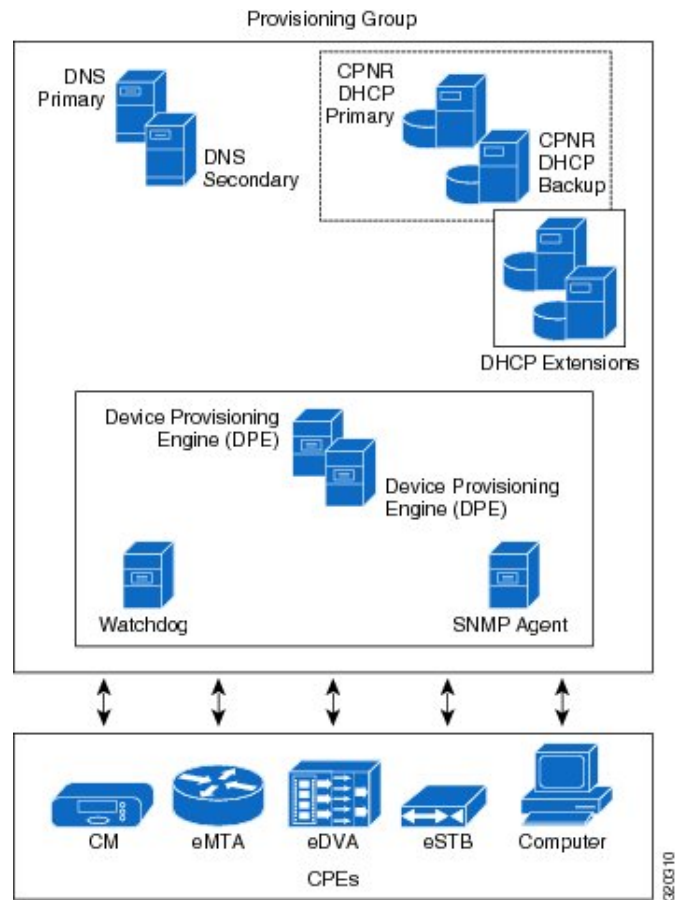The following figure represents a typical, fully redundant deployment in a Prime Cable Provisioning network.

Figure 8: Deployment Using Prime Cable Provisioning
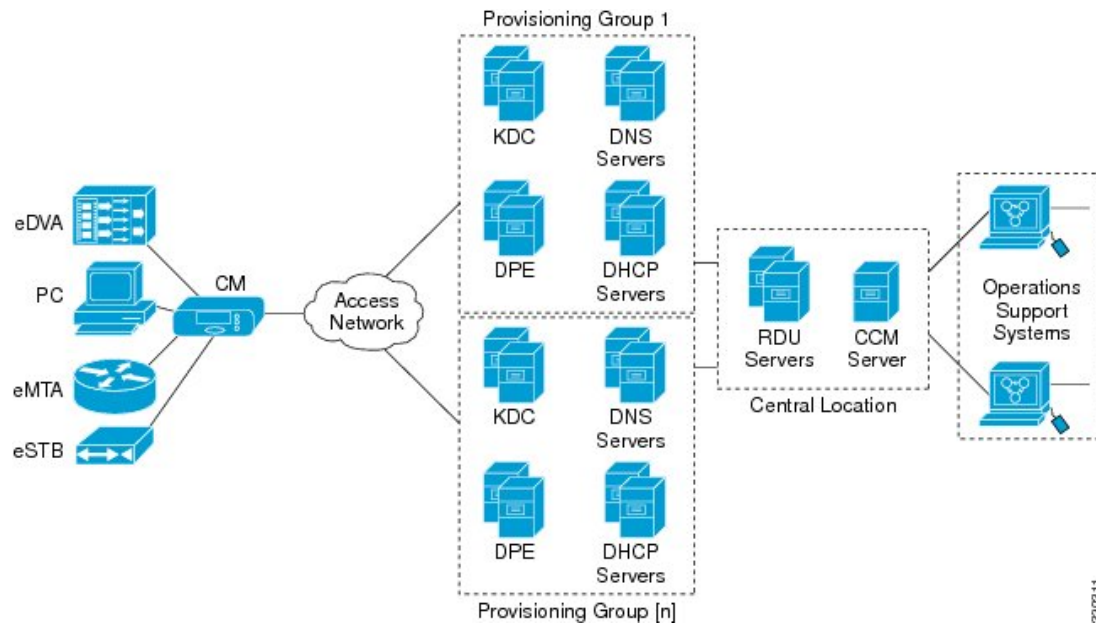
The following figure shows a typical provisioning group in a Prime Cable Provisioning network.

Figure 9: Typical Provisioning Group

The following figure represents a central location in a Prime Cable Provisioning network.

Figure 10: OSS in a Prime Cable Provisioning Network