Broadband Access Center for Cable Overview

Broadband Access Center for Cable (BACC) automates the configuration and provisioning of network devices supported by a broadband service provider. It is a flexible product that can be scaled to suit virtually any size network. BACC is designed to handle the rapid growth of the service provider. It targets broadband service providers (including multiple service operators), Internet, and voice service providers who want to deploy IP data, voice, and video on hybrid fiber and coaxial cable networks. BACC also provides such critical features as redundancy and failover protection, and can be integrated into new or existing environments through the use of a provisioning application programming interface (API) that lets you control how BACC operates. You can use the provisioning API to enable BACC to register devices, device configurations, and configure the entire BACC provisioning system.

Features and Benefits

BACC lets multiple service operators (MSOs) meet the rapidly changing demands for data over cable services. Using BACC, you can realize these features and benefits of its architecture:

- Increased scalability
- Distributed architecture
- Redundancy
- Extensibility

Supported Technologies

This BACC release supports these technologies:

- DOCSIS high speed data
- PacketCable voice services
- Non-secure CableHome provisioning
DOCSIS High Speed Data

The Data Over Cable Service Interface Specification (DOCSIS versions 1.0 and 1.1), defines functionality in cable modems involved in high-speed data distribution over cable television system networks. This allows MSOs to provide a variety of services through an “always-on” Internet connection. These services include broadband Internet connectivity, telephony, real-time interactive gaming, and video conferencing.

Note

Broadband Access Center for Cable supports DOCSIS 1.0 and 1.1 devices.

PacketCable Voice Services

PacketCable voice technology describes the delivery of advanced, real-time multimedia services over a two-way cable network. PacketCable is built on top of the infrastructure supported by cable modems to enable a wide range of multimedia services such as; IP telephony, multimedia conferencing, interactive gaming, and general multimedia applications.

The PacketCable voice technology allows the offering of additional services, for example, basic and extended telephony services, to be delivered more efficiently and cost-effectively, over the broadband cable access network.

Non-secure CableHome Provisioning

Non-secure CableHome provisioning, hereafter referred to as home networking technology, is built on top of the existing DOCSIS standard and supports a ‘plug and play’ environment for home connectivity of residential broadband services. This form of home networking technology encompasses a DOCSIS home access device that is considered to be the home’s entry point. The home access device is a combination cable modem and various features that are referred to as portal services which, when included in a single physical device, are known as embedded portal services.

Architecture

This section describes the basic BACC architecture including:

- Regional distribution unit (RDU) that provides:
  - The authoritative data store of the BACC system.
  - Support for processing API requests.
  - A device detection/configuration/disruption engine, which invokes the various installed extensions to determine device types, generate configurations, and support disruption for specific device types.

See the “Regional Distribution Unit” section on page 1-3 for additional information.

- Device provisioning engines (DPEs) that provide:
  - TFTP protocol server
  - Configuration cache
  - Redundancy
- Time of day server
- PacketCable provisioning services
See the “Device Provisioning Engines” section on page 1-4 for additional information.

- Provisioning Groups
See the “Provisioning Groups” section on page 1-7 for additional information.

- Cisco Network Registrar servers that provide:
  - Dynamic Host Configuration Protocol (DHCP)
  - Domain Name System (DNS)
See the “Cisco Network Registrar” section on page 1-6 for additional information.

- A Kerberos server that authenticates PacketCable MTAs. See the “KDC” section on page 1-8 for additional information.

- An SNMP agent that provides:
  - SNMP version v2c support
  - SNMP Trap support
See the “SNMP Agents” section on page 1-11 for additional information.

- The BACC agent that provides:
  - Administrative monitoring of all critical BACC processes.
  - Automated process restart capability.
  - The ability to start and stop BACC component processes. See the “BACC Agent” section on page 1-13 for additional information.

## Regional Distribution Unit

The RDU is the primary server in the BACC provisioning system. The RDU also provides:

- Management of device configuration generation.
- Authoritative datastore for the BACC system.
- Device disruption.
- A clearinghouse function, through which all application programming interface (API) requests must pass.
- Management of the BACC system.

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

- Provisioning data access and manipulation by means of the API
- Distributing configurations to the DPEs for scalability
- External clients, operations support systems (OSSs), and other provisioning related functionality by means of the API.

These sections describe these RDU concepts:

- Configuration Generation, page 1-4
- Regional Distribution Unit Failover, page 1-4
Configuration Generation

When a device boots, it requests a configuration from BACC and this configuration determines the level of service for the device. Device configurations can include customer required provisioning information, such as: DHCP IP address selection, bandwidth, data rates, flow control, communication speeds, and level of service. A configuration can contain DHCP configuration and TFTP files for any device. When an unprovisioned device is installed, and the boot operation performed, a default configuration for the appropriate technology is obtained from BACC and sent to the device, by means of either DHCP or TFTP. The default configuration can be changed for each supported technology.

Regional Distribution Unit Failover

Broadband Access Center for Cable currently supports one RDU per installation. For failover support, use your own hardware redundant system or obtain a similar system with hot-swap capability.

Device Provisioning Engines

The Cisco Device Provisioning Engine (DPE) communicates with the RDU to give devices their configurations. Each DPE caches information for up to 500 thousand devices and multiple DPEs can be used to ensure redundancy and scalability. Two types of DPE are supported by this BACC release; the traditional hardware DPE-590 device and the software only Solaris DPE. See the “Hardware Device Provisioning Engines” section on page 1-5 and the “Solaris Device Provisioning Engines” section on page 1-5 for additional information.

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

The DPE manages these activities:

- Last step device configuration generation (DOCSIS timestamps for instance)
- Communication of the configuration files through an embedded TFTP server
- Integration with Cisco Network Registrar
- Time of day protocol server
- Voice technology provisioning services

In addition to the different types of DPE, this section describes these major DPE components:

- “DPE License Keys” section on page 1-5
- “DPE Assignments” section on page 1-6
- “TFTP Server” section on page 1-6
- “Provisioning Groups” section on page 1-7
Hardware Device Provisioning Engines

BACC currently supports the DPE-590 hardware device provisioning engine.

**Note**
Refer to the *Cisco Content Engine 500 Series* product documentation or more information on the device itself. For specific information about the front and rear panel, the ports, and receptacles, see the *Cisco Content Engine 500 Series Site Preparation and Safety Guide*.

Solaris Device Provisioning Engines

The Solaris DPE functions in the same fashion as the hardware DPE with the exception that this part of the BACC product is installed on a computer running either the Solaris 8 or 9 operating system. With few exceptions, the same command line interface CLI is used on both the hardware and Solaris DPEs. See Table C-1 in Appendix C, “Device Provisioning Engine Command Line Interface” for specific information on which DPE supports which CLI command.

DPE License Keys

Licensing controls the number of DPEs (nodes) you can use. If you attempt to install more DPEs than you are licensed to have, those new DPEs will not be registered and, although they are rejected, existing valid DPEs remain online and re-register themselves with the RDU at will.

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

The number of DPE licenses you register with the RDU includes hardware and Solaris DPEs, regardless of release number or type; including those used as part of a BACC lab installation. See the *Broadband Access Center for Cable Installation Guide* for additional information.

Whenever you change licenses, either by adding a license, extending an evaluation license, or through the expiration of an evaluation license, the changes take immediate effect.

When you delete a DPE from the RDU database, a license is freed. The DPE automatically re-registers with the RDU when deleted.

Deleted DPEs are removed from all provisioning groups they serve and all Network Registrar extensions are notified that the DPE is no longer available. Consequently, when a previously deleted DPE is re-registered, it is considered to be licensed again and will remain 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 administrator’s user interface. You can only determine the license state by examining the DPE and RDU log files (dpe.log and rdu.log).

Any deleted/unlicensed hardware DPE, running either BPR 2.0.x or BAC 2.5, will constantly attempt to re-register with the RDU. This is normal behavior for devices running these versions of software and all rejected registrations are recorded in the RDU log and in the generation of SYSLOG alerts identifying this situation.
DPE Assignments

BACC supports multiple DPEs. These communicate with devices, a DHCP failover configuration, and the RDU. During installation, you must make these DPE assignments:

- Assign a DPE to be responsible for one or more logical groups of devices or provisioning groups.
- The IP address and port number of the RDU.

The DPE’s primary objective is to send configurations to customer devices whenever those devices are powered up or rebooted. To do this quickly, the DPE keeps a copy of the configuration for each device in a local cache database.

TFTP Server

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

The TFTP server is located in a home directory that is used for local file system access. The local files are stored in the BPR_DATA/dpe/tftp directory.

By default the TFTP server only looks in its cache for a TFTP read. However, if the tftp allow-read-access command has been run, the TFTP server first looks at 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 is there, the server uses it. Otherwise, the server sends a request to the RDU for the file. When you can enable read access from the local file system, directory structure read requests such as docsisfiles/gold.cm are allowed. This is only allowed from the local file system.

BACC only allows writes to the local file system; never to the DPE cache. If you run the tftp allow-write-access DPE CLI command, BACC will allow a write to the TFTP home directory. By default, the creation of directories or the override files is not permitted. However, to change this, run either the tftp allow-create-dirs or the tftp allow-override commands.

Cisco Network Registrar

Network Registrar provides Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) functionality. It has a complete administrative user interface that, when coupled with customized BACC configuration screens, can be used within a larger enterprise management system.

Note

For additional information on Network Registrar, refer to these documents: Network Registrar User's Guide, Network Registrar CLI Reference, and the Network Registrar Installation Guide.

Dynamic Host Configuration Protocol

The DHCP server automates the process of configuring IP addresses on TCP/IP networks. This 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 in itself, adds flexibility, mobility, and control to networked device configurations.
DHCP Failover

DHCP failover is a protocol designed to allow multiple servers to operate on a single network. Network Registrar allows a backup DHCP server to take over for a main Network registrar DHCP server, if that main server fails for any reason. For this to work reliably, and to prevent loss of access to the DHCP service, the cooperating primary and secondary servers maintain a consistent information database.

Domain Name System

The Domain Name System (DNS) is a server that contains information on hosts throughout the network, including IP address hostnames and routing information and DNS uses them primarily to translate between IP addresses and domain names. This conversion of names, such as www.cisco.com, to IP addresses simplifies accessing Internet-based applications. The DNS directory service consists of:

- DNS data
- DNS servers
- Internet protocols for fetching data from the servers
- Dynamic DNS for voice provisioning.

Provisioning Groups

A provisioning group is designed to be a regional grouping of servers usually consisting of a two or more DPEs and a failover pair of DHCP servers that (when using hardware DPEs) can handle the provisioning needs of up to 500 thousand devices. As the number of devices grows past 500 thousand, 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 NOC.

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 in a round-robin fashion so that any one DPE does not get overloaded with all the requests. The information stored on each DPE within a provisioning group is identical so as long as operational services are not interrupted. Figure 1-1 shows some typical provisioning group setups with varying numbers of DPEs.
The provisioning group at the bottom (C) shows the absolute minimum number of servers that can be used, one DPE and one DHCP. Cisco does not usually recommend that this be deployed as there is no redundancy or failover support and if one of the two servers stop functioning then all service is lost until it is back up again. This configuration could be used if the number of users being serviced in a particular region is so small that the additional hardware cost cannot be justified and the deployment is willing to live with an outage if a server stop functioning.

The provisioning group at the top (A) is the most commonly deployed one. A failover pair of DHCP servers and a redundant pair of DPEs. This provides enough load sharing and redundancy to support most regional deployments.

The provisioning group in the center (B) might be used in an area that has a higher constant rate of change to the devices being serviced or where large numbers of customer devices go up and down frequently. The addition of two DPEs allows the extra load to be shared over four DPEs, which allows greater numbers of devices to be serviced in the same amount of time or less.

**KDC**

The key distribution center (KDC) is an authentication server used to authenticate MTAs and grant security tickets to them.

**Note**

The KDC is supported on multi-processor computers.
Default KDC Properties

The KDC has several default properties that get populated, during BACC installation, into a `<BPR_HOME>/kdc/solaris/kdc.ini` properties file. This file can be edited to change values as operational requirements dictate. Once changes have been made, you must restart the KDC before any property file, key or certificate changes can take affect. The default properties are:

- **interface address**—This is the IP address of the local Ethernet interface that you want the KDC to monitor for incoming Kerberos messages. For example:

  ```bash
  interface address = 10.10.10.1
  ```

- **FQDN**—Identifies the fully qualified domain name (FQDN) on which the KDC is installed. For example:

  ```bash
  FQDN = kdc.cisco.com
  ```

  **Note:** The interface address and FQDN are entered through the KDC Realm Name screen during installation. Refer to the *Broadband Access Center for Cable Installation Guide* for specific information.

- **maximum log file size**—The KDC generates a set of log files. This property specifies the maximum size, in kilobytes, that the log file can reach. Therefore, the KDC will create a new log file only when the current file reaches this maximum size. For example:

  ```bash
  maximum log file size = 1000
  ```

- **n saved log files**—This property defines the number of old log files that the KDC saves. While the default value is 7, you can specify as many as required. For example:

  ```bash
  n saved log files = 10
  ```

- **log debug level**—This property specifies the logging level for the log file.

  ```bash
  log debug level = 5
  ```

- **minimum (maximum) ps backoff**—This property specifies the minimum (or maximum) time, in tenths of a second, that the KDC will wait for BACC to respond to the FQDN-REQUEST. For example:

  ```bash
  minimum ps backoff = 150
  ```

Using the example values shown above, a sample INI file might contain data similar to that shown in Example 1-1.

**Example 1-1 Sample KDC INI Configuration File**

```bash
interface address = 10.10.10.1
FQDN = kdc.cisco.com
maximum log file size = 1000
n saved log files = 10
log debug level = 5
minimum ps backoff = 150
maximum ps backoff = 300
```

You can set the times for both minimum and maximum ticket duration to effectively smooth out excessive numbers of ticket requests that could occur during deployment. This is beneficial given that most deployments occur during traditional working hours and excessive loading may, from time to time, adversely affect performance.
Note

Shortening the ticket duration forces the MTA to authenticate to the KDC much more frequently. Unfortunately, while this results in much greater control over the authorization of telephony endpoints, it also causes much heavier message loads on the KDC and increased network traffic. For most circumstances, the default setting is appropriate and should not be changed.

- maximum ticket duration—This property defines the maximum duration for tickets generated by the KDC. The default unit is hours; however, by appending an m or d, the units can be changed to minutes or days, respectively.

  The default value is 168, or seven days, and Cisco recommends that you not change this value since this is the duration required to conform to the PacketCable security specification. For example:
  
  \[\text{maximum ticket duration} = 168\]

- minimum ticket duration—This property defines the minimum duration for tickets generated by the KDC. The default unit is hours; however, by appending an m or d, the units can be changed to minutes or days, respectively.

  The default value is 144, or six days, and Cisco recommends that you not change this value. For example:
  
  \[\text{minimum ticket duration} = 144\]

KDC Certificates

The certificates used to authenticate the KDC, are not shipped with BACC. 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. See the “Managing KDC Certificates with the PKCert Tool” section on page 8-27 for additional information.

Warning

Without the certificates installed, the KDC will not function.

KDC Licenses

You obtain a KDC license from your Cisco representative and then you have to install it in the correct directory.

Note

To install a KDC licence file, copy it into the \(<\text{BPR\_HOME}>/\text{kdc}\) directory and name it \(\text{kdc.license}\). After doing so, run the \(\text{bprAgent restart kdc}\) command, from the \(/\text{etc/init.d}\) directory, to restart the KDC server and make the changes take effect.

To install a KDC license file:

Step 1 Obtain your license file.
Step 2 Copy that file to the \(<\text{BPR\_HOME}>/\text{kdc}\) directory.
Step 3 Rename the file to \text{kdc.license}.
Step 4 Run the \text{bprAgent restart kdc} command, from the \(/\text{etc/init.d}\) directory, to restart the KDC server and make the changes take effect.
Note
KDC license files should not be copied between operating systems, as the transfer process may damage the file. It is recommended that you compress the license file before transferring it.

SNMP Agents

BACC supports basic SNMP based monitoring of both the DPE and RDU servers.

All SNMP agents described in this section support the SNMP version v2c. The SNMP agents do not perform any logging. You can enable these agents using either the DPE CLI or RDU SNMP configuration CLI commands. See the Appendix C, “Device Provisioning Engine Command Line Interface” for additional information on the DPE CLI and the “Using the rduSnmpAgent.sh Command” section on page 8-32 for additional information on RDU SNMP configuration command line tool.

DPE SNMP Agent

The SNMP agent does not communicate directly with the BACC process. Instead, all feature status information is sent to the SNMP agent through the BACC agent. The DPE SNMP agent is also RFC-1213 compliant.

Note
The DPE SNMP agent is only supported on the hardware DPE.

MIB Support

The DPE SNMP agent supports these MIBs on the DPE-590:

- RFC 1213 (MIB-II)
- CISCO-CW-APPLIANCE_MIB

Note
These MIBs are located in the <BPR_HOME>/rdu/mibs directory.

SNMP Traps

SNMP traps are generated whenever the DPE process either goes up or down, and may also be generated if resource usage exceeds preset thresholds.

Table 1-1 identifies the SNMP traps that are generated. Note that some of these traps are triggered based on the resource usage level.
Table 1-1  DPE SNMP Agent Traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ciscoCwApplianceReadyToReload</td>
<td>This trap is generated whenever the DPE is reloaded by a CLI command. See the “reload” section on page C-20 for additional information.</td>
</tr>
<tr>
<td>ciscoCwApplianceReadyToShutdown</td>
<td>This trap is generated whenever a poweroff command is issued on the DPE CLI. See the “poweroff” section on page C-19 for additional information.</td>
</tr>
<tr>
<td>ciscoCwApplianceFeatureEnabled</td>
<td>This trap is generated whenever a feature on the DPE box is enabled. For example, this trap will be fired when the DPE feature on the DPE-590 is started. See the “DPE Configuration Commands” section on page C-34 for additional information.</td>
</tr>
<tr>
<td>ciscoCwApplianceFeatureDisabled</td>
<td>This trap is generated whenever a feature on the DPE is disabled. For example, this trap is fired when the DPE is stopped.</td>
</tr>
<tr>
<td>ciscoCwApplianceCPUBusy</td>
<td>This trap is generated whenever CPU utilization exceeds a pre-set threshold. CPU utilization is polled regularly and this trap is generated when the average of two consecutive sampling points is greater than the threshold value. Note This trap is generated as a result of monitoring resource use.</td>
</tr>
<tr>
<td>ciscoCwApplianceMemoryFull</td>
<td>This trap is generated when memory utilization exceeds a pre-set threshold. Memory utilization is polled regularly and this trap is generated when utilization exceeds a pre-set threshold. Note This trap is generated as a result of monitoring resource use.</td>
</tr>
<tr>
<td>ciscoCwApplianceDiskFull</td>
<td>This trap is generated whenever polling indicates that the amount of free hard disk space drops below the pre-set threshold level. Note This trap is generated as a result of monitoring resource use.</td>
</tr>
</tbody>
</table>

DPE Agent CLI

There are several SNMP related CLI commands that are necessary to use the agent. These are described in the “SNMP Agent Commands” section on page C-69.
RDU SNMP Agent

The RDU SNMP agent monitors the RDU process and generates SNMP traps whenever the RDU process starts or stops.

Since the RDU is intended to be loaded on an open Solaris device, the RDU SNMP agent will not provide support for other MIBs like RFC 1213 (MIB II).

MIBs Supported

The RDU SNMP agent is intended for operation within the Solaris operating system only and, as such, offers no support for RFC 1213 (MIB-II). This agent does however, support the CISCO-NMS-APPL-HEALTH-MIB which defines the Cisco NMS application health status notifications and related objects. These notifications are sent to the OSS/NMS to inform them about the NMS application status. Status can include: started, stopped, failed, busy, or any abnormal exit of applications.

Traps

The RDU SNMP agent generates this trap:
- cnaHealthNotif—This notification is sent out to announce BACC startup, shutdown, failed, or exit status. For example, this notification is sent out whenever the RDU server has started, shutdown or failed.

BACC Agent

The BACC agent is an administrative agent that monitors the run time health of all BACC processes. This watchdog process ensures that if a process stops unexpectedly, it is automatically restarted.

The BACC agent can be used as a command line tool to start, stop, restart, and determine the status of any monitored processes.

Monitored Processes

If a monitored application fails, it is restarted automatically. If, for any reason, the restart process also fails, the BACC agent server will wait a prescribed amount of time to attempt the restart again.

Note

You do not have to use the BACC agent to monitor Network Registrar extensions.

The period between restart attempts increases exponentially until it reaches a length of 5 minutes. After that, the process restart is attempted at 5 minute intervals until successful. Five minutes after a successful restart, the period is automatically reset to 1 second again.

For example:
- Process A fails.
- The BACC agent server attempts to restart it and the first restart fails.
- The BACC agent server waits 2 seconds and attempts to restart the process and the second restart fails.
The BACC agent server waits 4 seconds and attempts to restart the process and the third restart fails.

The BACC agent server waits 16 seconds and attempts to restart the process and the fourth restart fails.

**BACC Agent Command Line**

The BACC agent automatically starts whenever the system boots up. Consequently, this agent also starts those BACC system components it is configured to control. The BACC agent can also be controlled through a simple command line interface. This is performed by running the bprAgent command from the /etc/init.d directory.

Table 1-2 describes the CLI commands available for use with the BACC agent. You can run the CLI from the /etc/init.d directory.

**Table 1-2  BACC Command Line Interface**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bprAgent start</td>
<td>Starts the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent stop</td>
<td>Stops BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent restart</td>
<td>Restarts the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent status</td>
<td>Gets the status of the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent start &lt;process-name&gt;</td>
<td>Starts one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
<tr>
<td>bprAgent stop &lt;process-name&gt;</td>
<td>Stops one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
<tr>
<td>bprAgent restart &lt;process-name&gt;</td>
<td>Restarts one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
<tr>
<td>bprAgent status &lt;process-name&gt;</td>
<td>Gets the status of one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
</tbody>
</table>

**Note**  The <process-name> mentioned in Table 1-2 can be one of the rdu, kdc, dpe, rduSnmpAgent, and jrun (which runs the administrators and sample user interface) processes. The CLI process is also monitored in both Lab and DPE installations.

---

**Note**  The RDU operates in a Solaris environment and may not shut down satisfactorily each time the Solaris reboot command is used. The preferred command to bring down the system is shutdown.