Cuda 12000 IP Access Switch CLI-based Administration Guide

Release 3.0

PART NO. 780-000052-00
PUBLISHED SEPTEMBER 2001

http://www.adc.com

ADC Telecommunications, Inc.
8 Technology Drive
Westborough, MA 01581

The CUDA 12000 is listed to UL 1950 Third Edition and CAN/CSA-C22.2 No. 950-95 Third Edition compliance.

The following information is for compliance by Class A devices with FCC regulations: the equipment described in this manual has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC regulations. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

You can determine whether your equipment is causing interference by turning it off. If the interference stops, it was probably caused by the equipment or one of its peripheral devices. If the equipment causes interference to radio or television reception, try to correct the interference by using one or more of the following methods.

- Turn television or radio antenna until the interference stops.
- Move equipment to one side or the other of the television or radio.
- Move equipment farther away from the television or radio.
- Plug equipment into an outlet that is on a different circuit from the television or radio. *(That is, make certain the equipment and the television or radio are on circuits controlled by different circuit breakers or fuses.)*

Modifications to this equipment that are not authorized by ADC could void the FCC certification and UL approval and negate your authority to operate the equipment.

The Cuda 12000 includes RSA BSAFE cryptographic or security protocol software from RSA security. The Cuda 12000 contains an integrated DOCSIS-compliant provisioning server. Use of this provisioning functionality is restricted to licensed authorization. ADC will not support provisioning for for your use thereof if you are not authorized by the appropriate software license to use such provisioning.

The equipment and software described herein may be covered by an ADC warranty statement. You may obtain a copy of the applicable warranty by referring to www.adc.com/cable/support and selecting the *technical assistance* link. What follows is a summary of the warranty statement. The summary is not binding on ADC and is provided to you merely as a convenience.

The equipment warranty usually lasts twelve (12) months from point of shipment and the software warranty usually lasts sixty (60) days from the point of shipment. The software warranty covers both functionality as well as the media on which the software is delivered. Neither warranty entitles the customer to receive free and unlimited access for technical assistance. A separate technical support agreement must be purchased for unlimited access to technical support resources.

The equipment warranty only applies to the cost of a replacement component. It does not include the labor charge for installation of the replacement component. During the warranty period, warranty claims will be processed on a 10-day return to factory basis. Once the defective component is returned to the factory, ADC's sole liability under the equipment warranty shall be either:

- To repair or to replace, at ADC's option, the defective equipment component with a new or refurbished component; or
- If after repeated efforts ADC is unable to resolve the defect by repair or replacement, to refund the purchase price of the equipment or component upon return of the defective item.

A working component will be returned to the customer within 10 days after it is received by ADC.

The warranty period for repaired or replaced equipment components shall be the remainder of the original warranty period for the repaired or replaced item or ninety (90) days, whichever is greater.

Equipment warranty claims can be processed on-line through a web interface or directly by a customer support representative of ADC. As part of the standard process for issuing a Return Materials Authorization (RMA), the Customer Support organization will verify all reported failures prior to authorizing a shipment of a replacement part.

The equipment warranty does not cover any of the following events:

- The equipment has been subject to abnormal use, abnormal conditions, improper storage, exposure to moisture or dampness, unauthorized modifications, unauthorized connections, unauthorized repair, misuse, neglect, abuse, accident, alteration, improper installation, or other events which are not the fault of ADC, including damage caused by shipping;
- ADC or an authorized ADC distributor or reseller was not notified by the customer of the equipment defect during the applicable warranty period.

If the software media is unusable such that the software cannot be loaded onto the equipment, ADC will replace the media within 1 business day after ADC is notified through Technical Assistance Center.

During the software warranty period, ADC will provide software updates and/or maintenance releases at no additional charge to resolve any issues where the software does not function according to software specification. In order to receive on-going software maintenance releases after the 60-day warranty period, the customer must purchase the base level technical assistance agreement.

The software warranty does not cover any of the following events:

- Unauthorized modifications to the software or firmware;
- Unauthorized installation of non-ADC software on the Cuda 12000 platform;
- ADC or an authorized ADC distributor or reseller was not notified by the customer of the software defect during the applicable warranty period.

Non-ADC software may be warranted by its developer, owner or other authorized entity as expressly provided in the documentation accompanying such software.

Failures caused by non-ADC software are not covered by ADC's warranty and service activities to remedy such failures will be billed to the customer.

Remote technical assistance will be provided free of charge during the 60-day software warranty period. The hours for support during the warranty period are Monday through Friday from 8:00am to 5:00pm EST.

Additional hardware and software services are available by purchasing an extended service agreement. Contact your account representative or call 1-877-227-9783 for further details.

CONTENTS

13 CONFIGURING DHCP AUTHORITY

14 CONFIGURING IP

ABOUT THIS GUIDE

This chapter introduces you to the *Cuda 12000 IP Access Switch CLI-based Administration Guide* and contains the following sections:

Document Objective

The *Cuda 12000 IP Access Switch CLI-based Administration Guide* provides procedural information about the commands you can use to configure and manage the Cuda 12000 system using the command line interface (CLI). Before you use this guide, you should have already installed and brought the system online using the *Cuda 12000 IP Access Switch Installation Guide.*

The *Cuda 12000 IP Access Switch CLI-based Administration Guide* is a companion to the *Cuda 12000 IP Access Switch CLI Reference Guide,* which provides detailed reference information on CLI command syntax and arguments.

Audience

This guide targets the network administrator, responsible for configuring and managing the Cuda 12000 within a cable television headend site. It assumes a working knowledge of network operations, although it does not assume prior knowledge of ADC's network equipment.

Document Organization

The *Cuda 12000 IP Access Switch CLI-based Administration Guide* is organized as follows:

Part I: Administration Overview

Chapter 1: Cuda 12000 Overview — Provides an overview of product functionality and includes information on how the Cuda 12000 integrates into your network.

Chapter 2: About the Command Line Interface — Introduces you to the Cuda 12000 command line interface (CLI).

Chapter 3: Managing User Accounts — Provides information and procedures on how to create and configure user accounts for control of management access to the chassis.

Part II: Chassis Administration

Chapter 4: Chassis Configuration — Provides an overview of chassis-wide configuration and related tasks.

Chapter 5: Multi-Chassis Support — Provides information and procedures on how to create groups of Cuda 12000 chassis.

Chapter 6: Module Administration — Provides information and procedures for basic module administration, as well as Ethernet administration. Also includes information on how to view traffic statistics for each port.

Chapter 7: Packet Over SONET Administration — Provides information and procedures for Packet Over SONET administration.

Chapter 8: Timing and Alarm Controller Management — Describes the alarm management features that you can use to discover and troubleshoot cable modems, modules, and link problems. Also includes information on how to configure alarm reporting for attached fan tray and power supplies.

Chapter 9: Simple Network Management Protocol (SNMP) — Provides procedures for configuring the Cuda 12000 for SNMPv1, SNMPv2, and SNMPv3 management.

Chapter 10: Managing System Events — Describes how to manage event transmission and logging on the Cuda 12000.

Part III: IP Routing

Chapter 11: Creating Route Filters — Provides information and procedures for creating RIP and OSPF policy-based route filters.

Chapter 12: Configuring DHCP Relay — Provides information and procedures on how to configure DHCP relay on a cable interface.

Chapter 13: Configuring DHCP Authority — Provides information and procedures on how to configure DHCP authority on a cable interface.

Chapter 14: Configuring IP — Provides information and procedures on how to configure IP routing on your system. Includes information on Open Shortest Path First (OSPF) and Routing Information Protocol (RIP) configuration.

Chapter 15: IP Packet Filtering — Provides information and procedures for creating packet filters for cable interfaces.

Chapter 16: Network-Layer Bridging — Provides information and procedures for creating network-layer bridge groups. These bridge groups allow you to associate the same IP address with multiple system interfaces. A key value of this feature is the ability to span a single subnet across multiple system modules.

Chapter 17: Managing IP Multicast — Provides information and procedures for configuring the Cuda 12000 to route multicast traffic, which delivers a single stream of information to multiple destinations at one time. Includes information on IGMP and multicast routes.

Part IV: Cable Modem Termination Systems

Chapter 18: Configuring Cable Modem Termination Systems — Provides information and procedures for configuring and managing CMTS RF parameters. Provides instruction on the configuration of downstream and upstream channels, admission control, and advanced CMTS parameters.

Chapter 19: Managing Cable Modems — Provides information for managing and monitoring cable modems on the network.

Chapter 20: Subscriber Management — Describes how to configure subscriber traffic filtering and Customer Premise Equipment (CPE) device management on the Cuda 12000.

Chapter 21: MIB Browsing — Provides information on how to browse cable modem and MTA MIBs and the MIB objects that are returned.

Appendices

Appendix A: Command Summary — Provides a complete listing of CLI commands and a brief description of each; organized by function.

Appendix B: Configuring External Provisioning Servers — Provides information on configuring external FastFlow BPM and third-party provisioning servers.

Appendix C: Glossary — Provides a glossary of networking terms.

Notations

Table 1 lists the text notations that are used throughout the Cuda 12000 documentation set guide.

Table 1 Notice Conventions

Icon	Notice Type	Description
	Information Note	Important or useful information, such as features or instructions
	Caution	Information that alerts you to potential damage to the system
	Warning	Information that alerts you to potential personal injury

Command Syntax

Table 2 describes the command syntax conventions used in this guide.

Table 2 Command Syntax Conventions

Command Element	Syntax
Commands and keywords	Expressed in **bold**. For example: **show chassis-config**
Variables	Enclosed in < > and expressed in plain text. For example: **add arp** <ip-address> <mac-address> In this example, <ip-address> and <mac-address> are variables that follow the **add arp** command.
Optional Arguments	Enclosed in [ ]. For example: **ip route default** <gateway-ip-address> [<metric>] In this example, the variable <metric> is an optional argument.
Set of Choices	Enclosed in { | }. For example: **loop** {**line** | **internal**} In this example, the user can specify either the **line** keyword or the **internal** keyword following the **loop** command.
List	Expressed as three dots (...). For example: **snmp-server host** [<**notification-type**>...] In this example, the user can specify multiple notification types.

In examples only, all user input — commands, keywords, and variables — are in bold to distinguish what the user enters from display-only screen text. In all other sections of this document, the conventions described above apply.

Related Documentation

For more information on the Cuda 12000 system, refer to the following publications:

- ***Cuda 12000 IP Access Switch Installation Guide***: Provides the information you need to install the system and bring it online. Includes a test procedure to ensure that the system is operational and can provision modems.
- ***Cuda 12000 IP Access Switch CLI Reference Guide***: Provides detailed reference information on CLI command syntax and arguments.
- ***Cuda 12000 IP Access Switch CudaView Administration Guide***: Contains procedural information you need to configure and manage the system using CudaView.

Contacting Customer Support

To help you resolve any issues that you may encounter when installing, maintaining, and operating the Cuda 12000 system, you can reach Customer Support as follows:

- Phone: (877) 227-9783 (option 4)
- E-mail: support@basystems.com
- Customer Support Web Site — To access Customer Support on the Web, go to `http://www.adc.com/cable/support`, then select the *Technical Assistance Center* link. You can then report the problem online, search the ADC Customer Support database for known problems and solutions, and check *Frequently Asked Questions*.

When contacting Customer Support for technical assistance, be sure to have the following information ready:

- List of system hardware and software components, including revision levels and serial numbers.
- Diagnostic error messages.
- Details about recent system configuration changes, if applicable.

I ADMINISTRATION OVERVIEW

1 CUDA 12000 OVERVIEW

This chapter explains the overall features of the Cuda 12000 IP Access Switch and describes how your Cuda 12000 IP Access Switch fits into your network. This chapter consists of the following sections:

Introducing the Cuda 12000 IP Access Switch

The Cuda 12000 IP Access Switch is a fully-meshed IP access switch that sits between the hybrid fiber coax cables (HFC) and the carrier's IP backbone network. It serves as an integrated Cable Modem Termination System (CMTS) and IP router, and supports DOCSIS and EuroDOCSIS RFI Specification 1.0 and 1.1.

To understand the Cuda 12000 IP Access Switch, you need to understand the following aspects of the switch:

- Hardware
- Software
- Minimum Chassis Configuration

Hardware

This section provides a brief overview of Cuda 12000 IP Access Switch hardware features and modules. For more information on Cuda 12000 IP Access Switch hardware, refer to the *Cuda 12000 IP Access Switch Installation Guide*.

Features

The Cuda 12000 provides the following hardware features:

Table 1-1 Cuda 12000 Hardware Features

Feature	Description
Total System Redundancy	The entire system is architected for full redundancy to provide a highly fault-tolerant solution that includes: ■ Dual-Power Sources: The system can be connected to two -48 VDC power sources to ensure uninterrupted power availability. ■ MeshFlow™ Fabric: Every application module is connected to every other application module via a high-speed serial mesh. This mesh supports a peak throughput capacity of 204.6 Gbps. (132 x 1.55 Gbps.), delivering IP packet routing with minimal latency and high availability to guarantee Quality of Service (QoS) across your core IP network. ■ Dual Management modules: The Cuda 12000 supports up to two Management modules to ensure uninterrupted system management. ■ Redundant Management Buses: The backplane consists of a 100-Mbps management BUS with redundant channels, over which the Management modules and system application modules communicate.
Distributed Processing Power	Application modules consist of a network processor with dedicated Synchronous Burst SRAM. Unlike other systems that use a central system processor, processing power and memory scale with every application module that you install in the chassis.

Feature	Description
CableOnce™ Network Connections	The system supports a *CableOnce* design that allows you to cable directly to the appropriate connector fixed to the rear of the chassis, or slot backplate. Cabling directly to these stationary connectors, instead of to the modules themselves, allows module replacement without recabling. You remove a module and then insert a new one while the cables remain attached to the system. This blind-mate design also lets you pre-cable chassis slots to prepare them in advance for module installation at a later time.
Hot-swappable Modules	All system modules can be replaced while the system is running without interruption to other interconnected networks. Both application modules and Management modules are hot-swappable.

Modules

The Cuda 12000 IP Access Switch chassis comprises 14 slots. Twelve of the slots are for application modules and two of the slots are for management modules, which control the operations of the chassis. The following is a list of the modules supported by the Cuda 12000 IP Access Switch:

- Management Module
- DOCSIS Modules
  - 1x4 DOCSIS Module
  - 1x4 DOCSIS SpectraFlow Module
  - 1x6 DOCSIS SpectraFlow Module with Spectrum Management
- EuroDOCSIS Modules
  - 1x4 EuroDOCSIS Module
  - 1x4 EuroDOCSIS SpectraFlow Module
  - 1x4 EuroDOCSIS SpectraFlow Module with Spectrum Management
- Egress Modules (Route Server Modules)
  - Octal 10/100 Ethernet SpectraFlow Module
  - Gigabit Ethernet SpectraFlow Module
  - Packet over SONET (POS) SpectraFlow Module

DOCSIS (Data Over Cable Service Interface Specification) is a CableLabs® standard for interoperability between a CMTS and cable modems. EuroDOCSIS (European Data Over Cable Service Interface Specification) is a CableLabs® and tComLabs® standard.

DOCSIS and EuroDOCSIS modules serve as CMTS interface modules with your HFC network using upstream and downstream ports. Upstream ports support both QPSK and 16 QAM modulation; the downstream port supports 64/256 QAM modulation. Each application module has an independent network processor and Synchronous Burst RAM. As a result, processing power and memory scale with every module that you install in the chassis.

The route server module functions in a dual role as both a forwarding device and a route server. The configured route server module is an egress (non-DOCSIS) module, such as an Octal 10/100 Ethernet SpectraFlow Module, Gigabit Ethernet SpectraFlow Module, or Packet over SONET (POS) SpectraFlow Module.

While maintaining its original role as a forwarding module, the route server maintains a central routing table. This module then distributes the routing table to other application modules upon initialization, and incrementally updates the forwarding tables as new routes are discovered. Distributed forwarding tables on each application module provide an added level of fault tolerance; should the Management module or another application module fail, the existing operational modules forward traffic without interruption.

Software

The Cuda 12000 IP Access Switch system software comprises two software components, as follows:

- Base System Software (required): The base system software is shipped with your Cuda and contains the operating system. The base software includes the command line interface (CLI) and provides you with the following functions:
  - User Account Management
  - Chassis Configuration
  - Multi-Chassis Support
  - Module Administration
  - Event Management
  - SNMP Management
  - IP Configuration
  - Packet and Route Filter Creation
  - DHCP Relay Configuration
  - CMTS Administration
  - Cable Modem Administration
  - Subscriber Management
- CudaView (optional): CudaView contains the graphical user interface (GUI) component of the element management system. CudaView provides full management functionality in graphical views that are easy for users to understand. CudaView provides an intuitive and "top-down" visual display that accelerates the management learning curve for new users and improves the productivity of all users. CudaView offers topology views, fault views, performance graphs, and many other useful features. For more information on CudaView, refer to the *Cuda 12000 IP Access Switch CudaView Administration Guide.*

Minimum Chassis Configuration

The minimum configuration of a Cuda 12000 IP Access Switch comprises the following:

- A minimum of one management module, plus the base software package. The module and base software are required to configure the Cuda 12000 IP Access Switch.
- An Octal 10/100 Ethernet, Gigabit Ethernet, or POS module. Each of these modules offers these services:
  - A link from the Cuda 12000 to your network backbone
  - May be configured as the route server
  - May function in a dual forwarding role
- One DOCSIS or EuroDOCSIS application module, which is required to perform CMTS functions.

Understanding the Cuda 12000 Within Your Network

Cuda 12000 IP Access Switches are installed at the HFC end of a cable plant and are responsible for gateway operations between the headend and the Internet. Through the Cuda 12000 IP Access Switch, digital data signals are modulated onto RF channels for broadcast over the same infrastructure.

Typically, the signals are broadcast through the HFC to fiber nodes in the network. Amplifiers, coaxial cable, and taps carry the signals to the subscriber premises.

This example shows how the Cuda 12000 IP Access Switch can fit into your network.

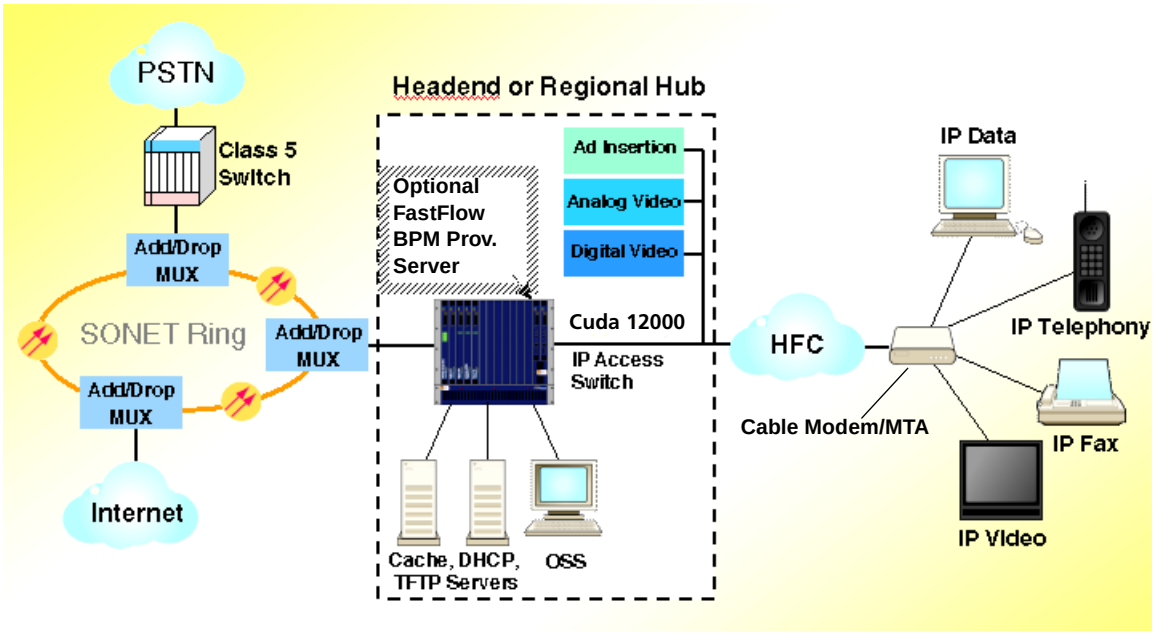


Figure 1-1 How the Cuda 12000 IP Access Switch Fits into Your Network

Cable Modem Termination System (CMTS)

The Cuda 12000 implements DOCSIS and EuroDOCSIS CMTS functionality, providing connectivity and data forwarding for cable modems over the RF cable plant.

The DOCSIS and EuroDOCSIS modules interface with your HFC network, using a 1-to-4 downstream-to-upstream port ratio (referred to as 1 x 4), or a 1-to-6 downstream-to-upstream port ratio (referred to as 1 x 6). Upstream ports support QPSK and 16 QAM modulation; the downstream port supports 64 and 256 QAM modulation.

IP Routing Configuration

The Cuda 12000 IP Access Switch uses the Internet Protocol (IP) to exchange data over computer networks consisting of cable and Ethernet interfaces. In addition, it supports RIP and OSPF routing protocols in order to exchange routing information with other routers in the IP network.

In order to integrate the Cuda 12000 IP Access Switch into your network, the following configuration must be accomplished:

- Configure the CMTS interfaces so that the cable modems range properly.
- Provision cable modems, Multimedia Terminal Adapters (MTAs), and CPE (Customer Premise Equipment) devices, using the FastFlow Broadband Provisioning Manager or a third-party provisioning server.
- Configure DHCP subnets, so that the DHCP server gives out IP addresses to cable modems, MTAs, and CPE devices.
- Configure IP on your cable, Ethernet, and Packet Over SONET interfaces to connect the Cuda 12000 to your backbone network and provide the subscribers access to the Internet.
- For the HFC segments, configure DHCP relay to specify the subnet to be used for assigning IP addresses to cable modems, MTAs, and CPE devices.

IP, RIP and OSPF can currently be configured on any of the interfaces within the Cuda12000 IP Access Switch.

2 About the Command Line Interface

This chapter introduces you to the command line interface (CLI) and covers the following topics:

About the CLI

The Cuda 12000 management module runs the Linux operating system. The CLI operates within this environment. The CLI is a textual command line interface accessible through a local COM port or through remote Telnet or secure shell (SSH).

The CLI operates within the command shell and offers a number of features to facilitate ease-of-use and configuration, including:

- **Context Sensitive Online Help** — The CLI offers the following online Help mechanisms:
  - **Individual Command Help** — You can display help on most commands by typing **help** followed by the command name. For example:

    ```
    cli:172.16.19.10:root# help bridge-group
    bridge-group                Creates a bridge group
        <name/id of bridge group>
    cli:172.16.19.10:root#
    ```

    The command name is listed on the left with a description on the right. Arguments are indented in standard syntax below the command name.

  - **Command Mode Help** — To view all commands available in the current mode with associated descriptions, type **help**. To show a list of available commands without descriptions, type **?** at the prompt or press the **Tab** key twice.

- **Configurable Prompt** — By default, the prompt displays both the address assigned to the management module and the current command mode. You can configure the prompt so it does not display this information. When the address and mode is displayed in the prompt, you can issue **set prompt** to remove it. To configure the prompt to display address and mode information, issue **set prompt mode**. For example:

  ```
  cli:172.16.19.10:root# set prompt
  cli#
  cli# set prompt mode
  cli:172.16.19.10:root#
  ```

- **Command Completion** — The system does not require that you type the entire command string. You simply need to type enough of the string to make it unique among the available commands so the system can recognize it. Once you type enough of the command string to distinguish it among other commands, simply press `[Tab]` to complete the command, or press `[Enter]` to execute it.

For most commands within the CLI, hyphens are placed between nouns, (such as **cpe-control***),* while no hyphen is placed between verbs and nouns *(*such as **no shutdown** and **show ip***)*. Also note that commands and their associated arguments are case-sensitive.

Accessing the CLI

Your first form of access to the CLI (after installing the Cuda 12000) is through COM port 1 located on the front of the management module. Once you assign the Craft Ethernet port on the management module an IP address, you can access the CLI remotely through Telnet or SSH.

Use the following procedure to logon to the system management module and access the CLI environment through COM port 1:

1. Ensure that you have cabled a console or a system running a terminal emulation program to COM port 1 and configured the correct serial transmission settings (57,600, 8, 1).
2. Access the system through the terminal emulator. Press [Enter] until you see the Linux login prompt.
3. You are then prompted for a login name and password to logon to the CLI. Enter your login name and password. The system ships with the following system defaults:
   - **Account Name:** administrator
   - **Password:** bas

   The Linux prompt is then displayed.

Note that the default login name and password are case-sensitive — all lowercase.

4. From the command prompt, access the CLI environment by issuing the following command:

   bascli
5. Within the CLI environment, enter your Cuda 12000 login name and password, as follows:

   enable <account name>

   <password>

   Note that the login name and password must be either defined locally on the Cuda 12000 or defined on a RADIUS or TACACS+ authentication server. Refer to Chapter 3 "Managing User Accounts" for more information on managing usernames and passwords.

The system ships with the following system defaults:

Account Name: root

Password: bas

For example:

```
[administrator@Tech2000 administrator]$ bascli
cli:null:root> enable root
password: ***
Connecting to 172.16.19.10...
Java Server version is compatible
ClientMode: CLI
logon complete
Sending message: User root just logged in from Tech2000

FROM:root@Tech2000:: User root just logged in from
Tech2000
```

Note that the default login name and password are case-sensitive — all lowercase.

Use the following procedure to logon to the system management module and access the CLI environment through the Craft Ethernet port:

1. Ensure that you have assigned an IP address to the Ethernet craft port on the management module, and that the Telnet and SSH server processes are running.
2. Open a Telnet session or an SSH session with the IP address or hostname assigned to the management module.
3. When the cli:null:root prompt appears, enter your Cuda 12000 login name and password, as follows:

   enable <account name>

   <password>

   The system ships with the following system defaults:

   Account Name: root

   Password: bas

For example:

```
ADC Cuda 12000

cli:null:root> enable root
password: ***
Connecting to 192.168.208.3...
Java Server version is compatible
logon complete
Sending message: User root just logged in from techpubs

FROM:root@techpubs:: User root just logged in from
techpubs
```

Note that the default login name and password are case-sensitive — all lowercase.

Command Modes

The Cuda 12000 switches and routes IP traffic between cable modems on an analog HFC network, and an IP digital network. As a result, administration tasks range from configuring IP interfaces and routing protocols to managing subscribers.

To support these administration tasks, the system provides a set of global commands and multiple command modes.

Global commands can be accessed anywhere in the CLI, while each command mode provides access to a set of related commands that cover a particular configuration scope. The current command mode is displayed in the prompt by default; you can verify the current mode that you are in at anytime by using the **show mode** command.

Command mode structure follows a hierarchy in which some modes run within others; all run within root mode. You can back up to the parent level from any sub mode using the **up** command. For local access, note that you can exit the CLI command shell and return to the Linux prompt at any time by typing **quit.** For Telnet or SSH access, the **quit** command terminates your session (you can also type **q**, which is a shortened form of **quit**).

You can also display a list of all available commands within the current mode by using one of the following help commands:

Help Command	Description
help	Displays the commands and command descriptions.
?	Displays all commands available within the current mode without descriptions; you can also display all commands by pressing the **Tab** key twice.

The command modes that are available for system configuration depend on the product packages installed. Base package system management command modes include:

- Root Mode
- Physical Interface Mode
- IP Interface Mode
- OSPF Global Configuration Mode
- Import and Export OSPF Route Filter Modes
- RIP Configuration Mode
- Import and Export RIP Route Filter Modes
- Slot Mode

If the FastFlow Broadband Provisioning Manager is installed on your Cuda 12000 IP Access Switch, additional command modes are available. Refer to the *FastFlow Broadband Provisioning Manager CLI-based Administration Guide* and the *FastFlow Broadband Provisioning Manager CLI Reference Guide* for more information on FastFlow Broadband Provisioning Manager command modes.

Global Commands

Global commands can be used anywhere in the CLI, regardless of your current command mode. Table 2-1 lists global commands as they appear when you type **help** at the command prompt. Note that the **help** command output displays many commands in their abbreviated form.

Table 2-1 Global Commands

Command	Description
basmonitor	Starts the system monitor.
boot	Enables, disables, or reboots a module in an application slot.
clear	Clears a specified system resource (depending on the specified argument), such as ARP cache or statistics counters.
cm-filter-default	Sets the default cable modem filter values for the system.
cpe-control	Sets the default subscriber management values for the system.
connect	Connects you to another Cuda 12000 IP Access Switch chassis.
echo	Echoes a comment so that it displays.
enable	Enables a user session.
help	Displays CLI command help.
interface	Changes you to interface mode.
ip	Configures IP parameters for the system.
no	Specifies the no form of a command.
ping	Enables you to the send an ICMP echo request packet to a destination to determine if it is reachable.
prov-server	Changes you to provisioning server mode. This command is useful only if the FastFlow Broadband Provisioning Manager is installed on your Cuda 12000 IP Access Switch.
q	Shortened form of **quit**.
quit	Enables you to exit from the CLI.
root	Changes you to root mode.
router	Changes you to router mode.

Table 2-1 Global Commands

Command	Description
server	Shortened form of **prov-server**.
set	Sets several user session parameters.
show	Specifies the show form of a command, which provides a read-only view of configuration parameters and other information.
sleep	Delays the display of the CLI prompt for a specified number of seconds.
slot	Changes you to slot mode.
source	Executes a script file.
talk	Enables and disables sending of broadcast messages. This command also allows you to send a message.
traceroute	Traces an IP route from a source to a destination.
up	Moves you up one level in the command mode hierarchy.

Root Mode

Root is the top-level mode in the CLI administration console; all other modes run within this mode. From within root mode you can access second-level command modes, such as slot configuration mode. To enter root mode from within any configuration mode, type **root**.

Table 2-2 lists available root commands as they appear when you type **help** at the command prompt. Global commands are not listed and can be found in Table 2-1 on page 42. Note that the **help** command output displays many commands in their abbreviated form.

Table 2-2 Root Mode Commands

Command	Description
aaa	Configures TACACS+ and RADIUS network authentication.
access-list	Creates an access list, which consists of IP filtering rules.
access-profile	Creates an access profile for a user.
account	Creates user accounts.
alarm-throttle	Configures alarm delivery and threshold parameters.
aux-device	Configures hardware alarm and clocking parameters.
bridge-group	Creates a bridge group.
bridge-timeout	Configures timers for bridge group broadcast flows.
ccdown	Shuts down the control module.
chassis	Configures multi-chassis support parameters.
chassis-config	Configures local chassis parameters.
chassis-fault	Configures chassis alarms.
cm-filter	Creates a cable modem filter.
db-check	Validates provisioning database information.
db-connect	Configures access to the provisioning database.
event-config	Configures event reporting, throttling, and syslog parameters.
event-log	Empties the event log.
http-server	Starts and stops the Web server.
lookup	Controls the Jini lookup service on the chassis.

Table 2-2 Root Mode Commands

Command	Description
modulation-profile	Configures modulation profiles, which contain burst properties for upstream data channels.
privacy	Configures X.509 certificate parameters for BPI plus.
radius-server	Configures a RADIUS authentication server.
reset	Reboots a module.
save	Saves the system configuration for all slots to persistent storage.
snmp-server	Configures the SNMP agent.
tacacs-server	Configures the TACACS+ server.
traffic-relay	Configures traffic relay for processes (such as servers) on the chassis.

Physical Interface Mode

Physical interface mode allows for the administration of a specified interface, including interface-specific configuration and information displays. To enter this mode, you must specify the chassis/slot/port-number *(c/s/i)* combination that identifies the physical interface that you want to configure. After you enter this mode, all configuration that you perform pertains to the interface that you specified.

To enter interface configuration mode, type **interface** <c/s/i> from within any configuration mode. The CLI automatically displays the type of interface for the specific c/s/i. The following tables list interface commands by type:

- Table 2-3 — Lists available DOCSIS interface mode commands as they appear when you type **help** at the command prompt.
- Table 2-4 — Lists available Ethernet interface mode commands as they appear when you type **help** at the command prompt.
- Table 2-5 — Lists available POS interface mode commands as they appear when you type **help** at the command prompt.

Keep in mind that the **help** command output displays many commands in their abbreviated form. Also keep in mind that global commands are not listed in any of these tables and can be found in Table 2-1 on page 42.

NOTE: *The commands displayed via* ***help*****,** ***?*****,** *and through the double Tab action are relevant to the selected interface.*

Table 2-3 DOCSIS Interface Mode Commands

Command	Description
access-class	Applies an access list to the current interface.
access-list	Creates an access list, which consists of IP filtering rules.
admission-control	Enables and disables admission control.
analyzer	Enables the protocol analyzer.
arp	Sets the ARP timeout parameter.
bootp-policy	Defines BOOTP request policies.
cable	Optional prefix to commands in this mode.

Table 2-3 DOCSIS Interface Mode Commands

Command	Description
cm	First element in various cable modem and subscriber management commands, such as **cm modify active**, **cm reset**, and so on.
cm-filter	Creates a cable modem filter.
cm-offline	Configures several offline cable modem parameters for the current interface.
dhcp-authority	Adds a DHCP authority range. The command also enables and disables DHCP authority.
dhcp-policy	Configures several parameters in the DHCP packet and determines the list of servers to which the DHCP packet is sent.
dhcp-relay	Configures various DHCP relay agent options, such as the IP addresses of cable modem, CPE, and MTA gateways.
downstream	Configures the downstream channel.
flap-list	Controls the size of the flap list.
insertion-interval	Configures the modem insertion interval.
link-trap	Enables link traps for the interface.
map-timer	Configures the map timer interval.
modulation-profile	Configures modulation profiles, which contain burst properties for upstream data channels.
periodic-ranging-interval	Configures how the interface periodically invites modems to range.
plant-delay	Configures the estimated plant propagation delay.
pll-state	Configures the phase lock loop state for the interface.
privacy	Configures BPI plus parameters for the interface.
proxy-arp	Enable proxy ARP on the interface.
qos	Enables SNMP and cable modem access to the QoS tables on the CMTS.
ranging-attempts	Configures the number of times that a cable modem is invited to range before being removed from the system.
shared-secret	Configures a shared secret on the current CMTS interface.
shutdown	Enables you to administratively shut down an interface.

Table 2-3 DOCSIS Interface Mode Commands

Command	Description
spectrum-group	Configures spectrum group rules.
sync-interval	Configures the time interval between synchronization message transmissions on the downstream channel.
trace-log	Configures event logging for the interface.
ucd-interval	Configures the time interval between transmission of successive Upstream Channel Descriptor (UCD) messages for each upstream channel.
upstream	Configures upstream channels.

Table 2-4 Ethernet Interface Mode Commands

Command	Description
access-class	Applies an access list to the current interface.
access-list	Creates an access list, which consists of IP filtering rules.
add	Adds a static ARP entry for the current interface.
arp	Sets the ARP timeout parameter.
bootp-policy	Defines BOOTP request policies.
dhcp-authority	Adds a DHCP authority range. The command also enables and disables DHCP authority.
dhcp-policy	Configures several parameters in the DHCP packet and determines the list of servers to which the DHCP packet is sent.
dhcp-relay	Configures DHCP relay agent options.
duplex	Configures the duplex mode for the interface (full duplex, half duplex, or auto).
link-trap	Enables link traps for the interface.
negotiation	Configures an Ethernet port to automatically negotiate duplex and speed.
shutdown	Enables you to administratively shut down an interface.
speed	Configures the speed for an Ethernet port.

Table 2-5 POS Interface Mode Commands

Command	Description
access-class	Applies an access list to the current interface.
access-list	Creates an access list, which consists of IP filtering rules.
arp	Sets the ARP timeout parameter.
bootp-policy	Defines BOOTP request policies.
clock-source	Configures the SONET transmission clock source.
crc	Configures cyclic redundancy checking (CRC).
dhcp-authority	Adds a DHCP authority range. The command also enables and disables DHCP authority.
dhcp-policy	Configures several parameters in the DHCP packet and determines the list of servers to which the DHCP packet is sent.
dhcp-relay	Configures DHCP relay agent options.
link-trap	Enables link traps for the interface.
loop	Configures the current interface for loopback testing.
mtu	Configures the maximum transmission unit (MTU) for the current interface.
pos	Configures POS parameters.
ppp	Configures PPP parameters.
shutdown	Enables you to administratively shut down an interface.

IP Interface Mode

IP interface mode allows for the administration of a specified IP interface, including IP interface-specific configuration and information displays. To enter this mode, you must:

1. Enter physical interface mode for the physical interface associated with the IP interface.
2. Issue the **ip address** command. On the command line, you specify the IP address and network mask combination that identifies the IP interface.

In IP address mode, the following commands are available:

- All commands that are available in the associated physical interface mode (DOCSIS, Ethernet, or POS).
- Commands for configuring RIP and OSPF on the interface (**ip rip** commands and **ip ospf** commands).

OSPF Global Configuration Mode

OSPF commands allow you to configure global OSPF (Open Shortest Path First) parameters. The system supports OSPF version 2 as defined in RFC 1583.

OSPF global configuration mode allows you to enable the protocol on a system-wide basis, and set system-wide OSPF parameters — such as router ID — and default OSPF parameters.

All OSPF areas to which you want this system to belong must be configured within this mode. You then assign areas to OSPF-enabled IP interfaces on an individual basis within IP interface mode. For example, if you want three IP interfaces to belong to three separate areas, you must first define the three areas using the **ospf area** command within this configuration mode. You can then use the **ip ospf** command within the IP interface configuration mode to assign the interface to one of the areas.

You can also set OSPF cost and dead-interval on a per-area basis. Configuration on a per-IP-interface basis overrides the same values that you define in OSPF global configuration mode.

To enter OSPF global configuration mode, type **router ospf** from any mode, or type **ospf** from router mode.

Table 2-6 lists available OSPF global commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 2-1 on page 42. Note that the **help** command output displays many commands in their abbreviated form.

Table 2-6 OSPF Global Configuration Mode Commands

Command	Description
asbr	Configures the Cuda 12000 IP Access Switch as an Autonomous System Boundary Router (ASBR).
export	Changes you to router export mode.
import	Changes you to router import mode.
ospf	Configures an OSPF area.
ospf-vi	Configures an OSPF virtual interface.
report	Enables sending of OSPF neighbor state and OSPF virtual neighbor state events.
router-id	Configures the OSPF router ID.

Import and Export OSPF Route Filter Modes

Route filters control the flow of routes to and from the routing table. Import route filters control which routes are stored in the routing table. Export filters control which routes are advertised to other routers. You can define route filters to control the flow of both OSPF and RIP routes.

To create OSPF import route filters, enter import mode from within router:ospf mode, or type **router ospf import** from any mode. You can then use the available commands to create route filters to control which OSPF routes the system learns.

To create OSPF export route filters, enter export mode from within router:ospf mode or type **router ospf export** from any mode. You can then use the available commands to create route filters to control which OSPF networks the router advertises to other OSPF routers.

Table 2-7 lists available OSPF import and export route filter commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 2-1 on page 42. Note that the **help** command output displays many commands in their abbreviated form.

Table 2-7 OSPF Import and Export Route Filter Mode Commands

Command	Description
asbr	Configures the Cuda 12000 IP Access Switch as an Autonomous System Boundary Router.
map-list	Adds a route map to a map list.
match	Specifies criteria that is matched against route entries.
ospf	Configures an OSPF area.
ospf-vi	Configures an OSPF virtual interface.
override	Enables you to override values for import or export filters.
report	Enables sending of OSPF neighbor state and OSPF virtual neighbor state events.
route-map	Creates a route map.
router-id	Configures the OSPF router ID.

RIP Configuration Mode

RIP (Routing Information Protocol) is a broadcast-based protocol used by routers to update routing tables, which include information about the networks that are in their routing tables. The routing table is broadcast to the other routers on the network where RIP is configured over IP.

The Cuda 12000 supports RIP version 2 as defined in RFC 1724. The Cuda can interoperate in a network of both RIPv1 and RIPv2 routers. A network composed of RIPv1 and RIPv2 routers is useful in supporting the transition from older routers to newer routers supporting RIPv2.

In order to exchange RIP routes over an interface you must configure RIP over IP on that interface. After RIP is added to the interface, the Cuda 12000 begins to exchange RIP routes with adjacent RIP routers.

To enter RIP configuration mode, type **router rip** from any mode, or type **rip** from router mode.

RIP configuration mode allows you to enter RIP import and export route filter modes using the **import** and **export** commands. It does not allow you to set global parameters. RIP parameters are configured on a per-IP-interface basis within IP interface mode by means of the **ip rip** command.

Import and Export RIP Route Filter Modes

Route filters control the flow of routes to and from the routing table. Import route filters control which routes are stored in the routing table. Export filters control which routes are advertised to other routers. You can define route filters to control the flow of both OSPF and RIP routes.

To create RIP import route filters, enter import mode from within router:rip mode or type **router rip import** from within any mode. You can then use the available commands to create route filters to control which RIP routes the system learns.

To create RIP export route filters, enter export mode from within router rip mode or type **router rip export** from within any mode. You can then use the available commands to create route filters to control which RIP networks the router advertises to other RIP-enabled routers.

Table 2-8 lists available RIP import and export route filter commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 2-1 on page 42. Note that the **help** command output displays many commands in their abbreviated form.

Table 2-8 RIP Import and Export Route Filter Mode Commands

Command	Description
map-list	Adds a route map to a map list.
match	Specifies criteria that is matched against route entries.
override	Configures the override values for import or export filters.
route-map	Creates a route map.

Slot Mode

Slot mode provides access to slot-specific commands. To enter this mode, you must specify a chassis/slot (c/s) combination that identifies the slot that you want to administer. Within this mode, you can do the following:

- Persist (save) configuration for the current module, or all modules in the system
- Configure and show trace log activity for the current slot
- Reset the module contained in the slot, or all modules in the chassis.

To enter slot mode, enter **slot** <chassis/slot> from within any mode. Table 2-9 lists available slot mode commands as they appear when you type **help** at the command prompt. CLI global commands are not listed and can be found in Table 2-1 on page 42. Note that the **help** command output displays many commands in their abbreviated form.

Table 2-9 Slot Mode Commands

Command	Description
copy	Downloads a file from a TFTP server to flash.
cpu-utilization	Enables CPU utilization on the module.
filter-aging	Configures IP packet filtering for all interfaces in the slot.
reset	Reboots a module.
save	Saves the system configuration for all slots to persistent storage.
trace-log	Configures event logging for the slot.

3 MANAGING USER ACCOUNTS

This chapter provides information and procedures on how to manage user accounts and consists of the following tasks:

- Configuring Access Profiles (page 57)
- Managing User Accounts (page 58)
- Configuring User Authentication (page 63)

Before you can effectively perform these tasks, you need to understand some concepts of user accounts.

Understanding User Accounts

You can manage security and control access to the system by creating and managing user accounts on the Cuda 12000. User accounts define both the functional areas the user can access and the types of access allowed for those areas.

In addition to creating user accounts locally on the Cuda 12000, you can also create user accounts on a TACACS+ or RADIUS authentication server. Refer to "Configuring User Authentication" on page 67 for more information.

Creating local accounts involves assigning access profiles to users. The accounts themselves are created using the **account** command. The access profiles that you assign to the account are created using the **access-profile** command.

You must have root profile privileges, as defined below, to manage user accounts.

Configuring Access Profiles

Access profiles define the type of access available to users. The **access profile** command allows you to configure access to the following functional areas:

Table 3-1 Functional Areas

Functional Area	Description
Admin	Functions associated with administering user accounts, such as adding modifying, and deleting users and profiles, as well as chassis configuration.
HFC	Functions associated with configuring and monitoring DOCSIS/EuroDOCSIS-related (CMTS) parameters such as configuring upstream and downstream channels.
Observer	Functions associated with a limited command set. The user has access to root mode and slot mode only, and is restricted to a limited set of commands. The user can type **help** or **?** to determine the available commands.
Prov	Functions associated with provisioning-related tasks, such as configuring DHCP servers and subnets.
Router	Functions associated with router-related tasks, such as configuring IP, RIP, and OSPF interfaces.

For each functional area, you can provide the following privileges:

- **noaccess**. Prevents the user from viewing or configuring the functional area.
- **readonly**. The user can view information for the functional area, but not configure it.
- **read/write**. The user can both configure and view the functional area.

The system ships with the following default access profiles. Note that these profiles are displayed in all capital letters when viewed to distinguish them from user-defined profiles. Also note that you cannot modify or remove these built-in profiles:

- **AUDITORPROFILE**. Provides read-only access to DOCSIS, routing, and provisioning functionality; no access to administrative functions.
- **OPERATORPROFILE**. Provides read-write access to DOCSIS, routing, and provisioning functionality; no access to administrative functions.
- **ROOTPROFILE**. Provides read-write access to all functional areas, including DOCSIS, routing, provisioning, and administrative functions.
- **NOACCESSPROFILE**. Denies access to all functional areas.

You can add more than one access profile to a user account. When you do so, the more powerful privileges take precedence. For example, if you assign both the AUDITORPROFILE and the ROOTPROFILE to a single user account, the ROOTPROFILE overrides the AUDITORPROFILE and the user has read-write access to all four functional product areas.

Creating and Modifying Access Profiles

To create or modify an access profile, use the **access-profile** command. To create a profile, specify a new profile name; to modify a profile specify an existing profile name. You create or modify a profile by performing the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Define the access profile.	**access-profile** <profile name> **description** <text string> {**addprivilege** | **removeprivilege**} {**admin** | **hfc** | **observer** | **prov** | **router**} {**noaccess** | **read/write** | **readonly**}

Example

The following example creates a profile with read only rights to routing functionality:

```
cli:172.16.19.10:root# access-profile routemonitor description
"Readonly" addprivilege router readonly
  PROFILE AFTER CREATE
    profileName: routemonitor
    profileDescription: Readonly
    PrivilegeList:
      router: readonly
'routemonitor' was successfully created
cli:172.16.19.10:root#
```

Displaying Access Profiles

You display access profile information by performing the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Display all access profiles.	**show access-profile**
3. Display a specific access profile.	**show access-profile** <profile name>

Example

The following example displays a profile named *routemonitor*:

```
cli:172.16.19.10:root# show access-profile routemonitor
Showing single profile:
    profileName: routemonitor
    profileDescription: Readonly
    PrivilegeList:
      router: readonly
cli:172.16.19.10:root#
```

Deleting a Profile

You remove an access profile by performing the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Remove a specified access profile.	**no access-profile** <profile name>

Example

The following example deletes an access profile named *routemonitor*:

```
cli:172.16.19.10:root# no access-profile routemonitor
'routemonitor' was successfully removed
cli:172.16.19.10:root#
```

Managing User Accounts

You create and modify local user accounts on the Cuda 12000 using the **account** command. You must have root profile privileges to manage user accounts which the following:

- Creating and Modifying User Accounts
- Displaying User Accounts
- Deleting User Accounts

For each user account, you define the following parameters:

Table 3-2 User Account Parameters

Parameter	Description
username	Name of the user as defined by the administrator that created the account.
password	Password required to access this account. Note that if you do not specify a password, a NULL value is added. This means that the user would simply press [Enter] to access the account.
description	Administrative description of the user account.
access-profile	Access profile that you want to assign to the user account. Note that you can add multiple profiles to a user account by reissuing the command.

Creating and Modifying User Accounts

To create or modify a user account, use the **account** command. To create a new account, specify a new account name. To modify an existing account, specify an existing name. You create or modify a user account by performing the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Create a user account.	**account** <account name> {**add-profile** <profile name> | **remove-profile** <profile name> | **password** <password> | **description** <string>}

Example

The following example creates a new user account named *Route_1* that uses the password *tech* and applies a profile named *routemonitor*:

```
cli:172.16.19.10:root# account Route_1 add-profile routemonitor
password tech description "Read Only Routing Admin"
  ACCOUNT AFTER CREATE
    UserName: Route_1
    UserDescription: Read Only Routing Admin
    PROFILE LIST
      profileName: routemonitor
      profileDescription: Readonly
      PrivilegeList:
         router: readonly

'Route_1' was successfully created
cli:172.16.19.10:root#
```

User account names and passwords are case-sensitive.

Displaying User Accounts

You view user accounts configured on the system by performing the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Show all user accounts.	**show account**
3. Show a specified user account.	**show account** <account name>

Example

The following example shows the user account named *Route_1*:

```
cli:172.16.19.10:root# show account Route_1
Showing single account:
    UserName: Route_1
    UserDescription: Read Only Routing Admin
    PROFILE LIST
      profileName: routemonitor
      profileDescription: Readonly
      PrivilegeList:
         router: readonly

cli:172.16.19.10:root#
```

Deleting User Accounts

You may want to delete a user account when you no longer need it or want to remove a user from the system. After a user account is deleted that user is locked out of the system.

Note that this is also true for the user with root profile privileges. If there is only one user with root profile privileges for your system and that user is locked out of the system, then you need to contact Customer Support for assistance.

To remove a user account from the system, perform the following task:

Task	Command
Delete account.	**no account** <account name>

Example

The following example removes the user account *Route_1* from the system:

```
cli:172.16.19.10:root# no account Route_1
'Route_1' was successfully removed
cli:172.16.19.10:root#
```

Configuring User Authentication

The Cuda 12000 IP Access Switch supports three types of user authentication:

- Local authentication – Users are authenticated locally by the Cuda 12000. This is the default authentication type.
- TACACS+ authentication – Users are authenticated by a TACACS+ server. When the user attempts to login to the Cuda 12000, the Cuda 12000 encrypts the username and pasword, and forwards them to the TACACS+ server for authentication. TACACS+ users are assigned the ROOTPROFILE access profile.
- RADIUS authentication – Users are authenticated by a RADIUS server. When the user attempts to login to the Cuda 12000, the Cuda 12000 encrypts the password only, and forwards the username and password to the RADIUS server for authentication. RADIUS users are assigned the ROOTPROFILE access profile.

The sections that follow describe how to configure user authentication.

Configuring Local Authentication

By default, users are authenticated locally by the Cuda 12000 using the accounts and access profiles you create as described earlier in this chapter. If you configure TACACS+ or RADIUS authentication, and then decide to change back to local authentication, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Enable local authentication.	**aaa authentication login default local**
3. Verify that local authentication is enabled.	**show aaa**

Example

```
cli:192.168.208.3:root# aaa authentication login default local
cli:192.168.208.3:root# show aaa
aaa authentication login default local
```

Configuring TACACS+ Authentication

Before you configure TACACS+ authentication on the Cuda 12000, make sure that:

- At least one account for Cuda 12000 users has been created on the TACACS+ server. Users must login to the Cuda 12000 an account created on the TACACS+ server. Refer to your TACACS+ server documentation for more information.
- You know the IP address of the TACACS+ server.
- You know the shared key that the Cuda 12000 will use to encrypt TACACS+ usernames and passwords for transmission to the TACACS+ server.

If TACACS+ authentication is unavailable due to problems with the TACACS+ server, local authentication is used.

To configure TACACS+ authentication on the Cuda 12000, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Specify the IP address of the TACACS+ server.	**tacacs-server host** <ip-address>
3. Specify the encryption key that the Cuda 12000 will use to encrypt usernames and passwords. The key is an alphanumeric string.	**tacacs-server key** <string>
4. Verify TACACS+ server settings.	**show tacacs-server**
5. Enable TACACS+ authentication.	**aaa authentication login default tacacs+**

Task	Command
6. Verify that TACACS+ authentication is enabled.	**show aaa**

Example

```
cli:192.168.208.3:root# tacacs-server host 192.168.220.200
cli:192.168.208.3:root# tacacs-server key one4me
cli:192.168.208.3:root# show tacacs-server
tacacs-server host 192.168.220.200
tacacs-server key one4me
cli:192.168.208.3:root# aaa authentication login default tacacs+
cli:192.168.208.3:root# show aaa
aaa authentication login default tacacs+
cli:192.168.208.3:root#
```

Configuring RADIUS Authentication

Before you configure RADIUS authentication on the Cuda 12000, make sure that:

- At least one account for Cuda 12000 users has been created on the RADIUS server. Users must login to the Cuda 12000 an account created on the RADIUS server. Refer to your RADIUS server documentation for more information.
- You know the IP address of the RADIUS server.
- You know the shared key that the Cuda 12000 will use to encrypt RADIUS passwords for transmission to the RADIUS server. Usernames are not encrypted.

If RADIUS authentication is unavailable due to problems with the RADIUS server, local authentication is used.

To configure RADIUS authentication on the Cuda 12000, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Specify the IP address of the RADIUS server.	**radius-server host** <ip-address>
3. Specify the encryption key that the Cuda 12000 will use to encrypt passwords. The key is an alphanumeric string.	**radius-server key** <string>
4. Verify RADIUS server settings.	**show radius-server**
5. Enable RADIUS authentication.	**aaa authentication login default radius**

Task	Command
6. Verify that RADIUS authentication is enabled.	**show aaa**

Example

```
cli:192.168.208.3:root# radius-server host 192.168.220.202
cli:192.168.208.3:root# radius-server key one4me
cli:192.168.208.3:root# show radius-server
radius-server host 192.168.220.202
radius-server key one4me
cli:192.168.208.3:root# aaa authentication login default radius
cli:192.168.208.3:root# show aaa
aaa authentication login default radius
cli:192.168.208.3:root#
```

II CHASSIS ADMINISTRATION

4 CHASSIS CONFIGURATION

This chapter explains the configuration features of the Cuda 12000 chassis and includes the following sections:

In addition to the features described in this chapter, you can group multiple chassis together to be managed through a single network management view. Refer to Chapter 5 "Multi-Chassis Support" for more information.

Understanding Chassis Identification

The Cuda 12000 chassis has two key identifiers:

- **Chassis Number** — The Cuda 12000 chassis is shipped with a unique chassis-number, which is a fixed value assigned to each chassis during manufacturing at the ADC plant.
- **Chassis ID** — Each Cuda 12000 switch should be configured with a unique chassis identification (ID) number. The chassis ID serves as a router management tool.

Understanding Management Module Redundancy

Each chassis is equipped with at least one management module, which controls the chassis. For management module redundancy, the Cuda 12000 supports installation of two management modules. When two management modules are installed, one acts as the *primary* management module and the other acts as the *secondary* management module.

When a Cuda 12000 that is configured with two management modules reboots, the Cuda randomly determines which module acts as the primary and which module acts as the secondary. The STATUS DISPLAY LED on the management module indicates whether the management module is a primary or secondary (for example, the LED on a primary management module displays "PRIMARY").

The primary management module is the active management module on the Cuda 12000. When you issue CLI commands or use the Cuda Desktop GUI to manage the Cuda 12000, you are interacting with the primary management module.

The secondary management module has two responsibilities:

- Monitor the state of the primary management module
- Keep its mirrored disk sectors synchronized with the primary management module

A secondary management module can take over the primary role in two ways:

- Automatically, when the secondary management module detects that the primary management module is not functioning properly.
- Manually, through the **chassis-config** CLI command. In this case, you use the command to force the current primary management module into the secondary role, which in turn forces the current secondary management module into the primary role.

When the secondary management module takes over the primary role, the secondary:

- Activates its copy of the chassis software
- Establishes connections with all other cards in the chassis

When the secondary management module activates its copy of the BAS software and establishes connections to cards, the secondary management module also starts services, including disk-mirroring and LDAP. Through disk mirroring, the software on the two management modules share data.

When a switch to a secondary management module occurs:

- Services are unavailable for a brief period of time
- Network management access is prevented for a brief period of time.

The Cuda's data-forwarding operation is not disrupted while a switchover to a backup occurs.

Configuring Chassis Parameters

Chassis configuration includes the following parameters:

- **Chassis Number** — A fixed number assigned to the Cuda 12000 chassis during manufacturing at the ADC plant.
- **Chassis ID —** User-defined. A unique identification number that you assign to the Cuda 12000 chassis in the network. The Cuda uses a multi-range numbering system. Acceptable chassis ID values are 1 to 128, or the number 255.
- **Cluster ID —** User-defined. An ID of "0" is the default and is recommended.
- **Manager** — Enables you to force the current primary management module into a secondary role, thereby forcing the current secondary management module into the primary role.
- **Scope** — Currently, "Cluster" is the only supported value.

Before you proceed to configure a chassis, you must know the number of the specific chassis. To display the current chassis configuration, including the current Chassis Number, use the **show chassis-config** command within root mode. For example:

```
cli:172.16.19.10:root# show chassis-config
Chassis Number:         101
Chassis Id:             1
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: 14
Scope:                  Cluster
cli:172.16.19.10:root#
```

Configuration of chassis parameters is achieved using the **chassis-config** command within root mode. Perform the following tasks within root mode to configure chassis parameters:

Task	Command
1. Identify the chassis number.	**show chassis-config**
2. Configure the chassis ID.	**chassis-config** <chassis-number> **chassisid** <1..128>
3. Configure the Cluster ID	**chassis-config** <chassis-number> **clusterid** <number>
4. Switch the primary management module to a secondary role, thereby forcing the secondary management module into the primary role.	**chassis-config** <chassis-number> **manager secondary**
5. Define the management scope of the primary or secondary chassis controller.	**chassis-config** <chassis-number> scope {**chassis** | **cluster**}

The following example steps you through the chassis configuration process:

```
cli:172.16.19.10:root# show chassis-config
Chassis Number:         101
Chassis Id:             5
Cluster Id:             20
Primary Manager Slot:   13
Secondary Manager Slot: 14
Scope:                  Chassis

cli:172.16.19.10:root# chassis-config 101 chassisid 1
cli:172.16.19.10:root# chassis-config 101 clusterid 0
cli:172.16.19.10:root# chassis-config 101 scope cluster
cli:172.16.19.10:root# show chassis-config
Chassis Number:         101
Chassis Id:             1
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: 14
Scope:                  Cluster

cli:172.16.19.10:root#
```

The following example shows you how to force a primary management module into a secondary role, thereby forcing the secondary management module into a primary role:

```
cli:192.168.222.200:root# show chassis-config
Chassis Number:         101
Chassis Id:             1
Cluster Id:             0
Primary Manager Slot:   13
Secondary Manager Slot: 14
Scope:                  Cluster
cli:192.168.222.200:root# chassis-config 101 manager secondary
Connection to 192.168.222.200 refused or closed!
```

Note that you must reconnect to the Cuda 12000 after you force the management modules to change roles.

Displaying Current Chassis Configuration

The Cuda 12000 allows you to generate a list of CLI commands that display the current running state of the chassis configuration. The command that supports this function is **show running-config**.

Use the following procedure to display the complete current configuration:

Task	Command
1. Enter **root** mode.	**root**
2. Display the current configuration.	**show running-config** [{**all** | **xml** | **server-name** <server-name>}]

Note that:

- When you issue the command with no arguments, the output does not display default values. Issue the command with the **all** argument to display all values.
- The **xml** argument displays the output in XML format.
- The **server-name** argument is reserved for ADC internal use only.

In addition to displaying the running configuration, you have the ability to generate a script to a file. This script may be used to configure other chassis to use the same configuration. You can write the script within the **bascli** environment using the **source** command; or you may use **-f** within a telnet session.

Example

```
cli:192.168.208.3:root# show running-config
!
!  BAS Chassis
event-config reporting warning local|syslog|traps
event-config reporting notice local|syslog|traps
event-config reporting info none
traffic-relay tftp port 69
traffic-relay time_of_day port 37
  !
  !  RIP Protocol
  router rip
    !
    !  RIP Import filters
    import
    up
    !
    !  RIP Export filters
    export
    up
  root
  !
  !  OSPF Protocol
  router ospf
  ospf area 0.0.0.0
    !
    !  OSPF Import filters
    import
    up
    !
    !  OSPF Export filters
    export
    up
  root
  !
  !  IP Protocol
  !
  !  Bridge Group Holder
  !
  !  Bas SNMP Manager
  snmp-server group adc v1 read public write private notify
public context adc s
torage nonvolatile
  snmp-server group adc v2c read public write private notify
public context adc
storage nonvolatile
```

```
  snmp-server group adc v3 noauth read public write private
notify public contex
t adc storage nonvolatile
  snmp-server group mgr v3 noauth read nosnmpconfig context
monitor storage nonv
olatile
  snmp-server group guitraps v1 notify guitraps storage readonly
  snmp-server group guitraps v2c notify guitraps storage
readonly
  snmp-server group superman v3 priv read allaccess write
allaccess context admi
n storage nonvolatile
  snmp-server group admingroup v1 read allaccess write allaccess
storage nonvola
tile
  snmp-server group admingroup v2c read allaccess write
allaccess storage nonvol
atile
  snmp-server group monitorgroup v1 read nosnmpconfig storage
nonvolatile
  snmp-server group monitorgroup v2c read nosnmpconfig storage
nonvolatile
  snmp-server group trapcommunity v1 notify allaccess storage
nonvolatile
  snmp-server group trapcommunity v2c notify allaccess storage
nonvolatile
  snmp-server view public 1.3.6.1 included storage nonvolatile
  snmp-server view private 1.3.6.1 included storage nonvolatile
  snmp-server view guitraps 1.3.6.1 included storage readonly
  snmp-server view allaccess 1.3.6.1 included storage
nonvolatile
  snmp-server view nosnmpconfig 1.3.6.1 included storage
nonvolatile
  snmp-server view nosnmpconfig 1.3.6.1.6.3 excluded storage
nonvolatile
  snmp-server community admincon admingroup address
192.168.212.0 mask 255.255.2
55.0 storage nonvolatile
  snmp-server community guitraps guitraps storage readonly
  snmp-server community justme admingroup address 100.100.10.5
address 100.100.1
0.8 address 192.168.212.109 storage nonvolatile
  snmp-server community monitor monitorgroup storage nonvolatile
  snmp-server community private adc context adc storage
nonvolatile
  snmp-server community public adc context adc storage
nonvolatile
```

```
  snmp-server community trapcommunity trapcommunity storage
nonvolatile
  snmp-server host 127.0.0.1 guitraps udp-port 54321 storage
readonly
  snmp-server host 201.1.1.20 trapcommunity version 1 udp-port
162
  !
  !  Fault Manager
  !
  !  Controller Module
  slot 1/13
  !
  !  Gigabit Ethernet Module
  slot 1/3
    !
    !  Interface 1/3/1 Gigabit Ethernet Mac
    interface ethernet 1/3/1
    root
  !
  !  10/100 Ethernet Module
  slot 1/11
    !
    !  Interface 1/11/1 10/100 Ethernet MAC
    interface ethernet 1/11/1
      !
      !  IP Address
      ip address 199.199.1.1 255.255.255.0
      ip ospf area-id 0.0.0.1
      up
    root
    !
    !  Interface 1/11/2 10/100 Ethernet MAC
    interface ethernet 1/11/2
    root
    !
    !  Interface 1/11/3 10/100 Ethernet MAC
    interface ethernet 1/11/3
    root
    !
    !  Interface 1/11/4 10/100 Ethernet MAC
    interface ethernet 1/11/4
    root
    !
    !  Interface 1/11/5 10/100 Ethernet MAC
    interface ethernet 1/11/5
    root
    !
    !  Interface 1/11/6 10/100 Ethernet MAC
```

```
    interface ethernet 1/11/6
    root
    !
    !  Interface 1/11/7 10/100 Ethernet MAC
    interface ethernet 1/11/7
    root
    !
    !  Interface 1/11/8 10/100 Ethernet MAC
    interface ethernet 1/11/8
    root
  !
  !  CMTS Module
  slot 1/1
    !
    !  Interface 1/1/1 CMTS MAC
    interface cable 1/1/1
    dhcp-policy default permit forward-internal
    dhcp-relay add-agent-options enable
    ip source-route 201.1.2.0 255.255.255.0 201.1.3.1
    ip source-route 201.1.4.0 255.255.255.0 201.1.5.0
    ip source-route 201.4.6.0 255.255.255.0 201.2.7.0
    admission-control disable
    privacy base auth-lifetime 604800 tek-lifetime 43200
cert-trust trusted enab
le-cert-validity-periods true
    cm-offline timer 10
    cpe-control max-ip 10
    cpe-control active
    trace-log baseline-privacy true registration true ranging
true
    !
    !  Interface 1/1/1 CMTS Downstream
    downstream transmit-power 0
    downstream no shutdown
    !
    !  Interface 1/1/1 CMTS Upstream 1
    upstream 1 frequency 42.0 voice-bw-reserve 65.0
    upstream 1 no shutdown
    !
    !  Interface 1/1/1 CMTS Upstream 2
    upstream 2 voice-bw-reserve 60.0
    upstream 2 no shutdown
    !
    !  Interface 1/1/1 CMTS Upstream 3
    !
    !  Interface 1/1/1 CMTS Upstream 4
  root
  !
```

```
!  POS Module
slot 1/8
  !
  !  Interface 1/8/1 POS MAC
  interface pos 1/8/1
  arp timeout 0
    !
    !  BasPppProtocol
  root
```

Configuring Clock Sources

The Cuda 12000 IP Access Switch backplane has a primary clock (A) and a secondary clock (B). For each of these clocks, you can configure one of the following clock sources:

- External BITS-A clock source
- External BITS-B clock source
- External Packet-Over-SONET (POS) clock source
- Internal Stratum-3 oscillator clock source on the management module

If you do not configure any clock sources, each DOCSIS/EuroDOCSIS module uses its own clock.

If you are using a BITS-A or BITS-B external clock source, make sure that the Cuda 12000 is connected to the appropriate clock sources via the BITS-A or BITS-B external clock connectors. Refer to the *Cuda 12000 IP Access Switch Installation Guide* for more information on these connectors.

If you are using a POS module as the clock source, make sure that the interface on the POS module has been configured to receive clocking from the other (remote) side of the POS link. To do this, issue the following command from within POS interface mode:

clock-source line

Refer to Chapter 7 "Packet Over SONET Administration" for more information on configuring POS interfaces.

A typical configuration would be as follows:

- Primary clock configured to use a BITS-A or BITS-B external clock source
- Secondary clock configured to use the internal Stratum-3 oscillator clock source.

The example at the end of this section illustrates the commands you would issue to create this typical configuration.

To configure primary and secondary clock sources, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Configure the clock source for the primary clock.	**aux-device backplane-clock-a** {**bits-a | bits-b | internal | none | slot** <c/s> {**enable | disable**}}
	Note that the **slot** <c/s> {**enable | disable**} *argument configures a POS module as the clock source. The* <c/s> *variable specifies the POS module's slot. Specify the* **enable** *keyword to enable the POS module as a clock source; specify the* **disable** *keyword to disable the POS module as a clock source.*
3. Configure the clock source for the secondary clock.	**aux-device backplane-clock-b** {**bits-a | bits-b | internal | none | slot** <c/s> {**enable | disable**}}
4. Verify primary and secondary clock sources.	**show aux-device backplane-clocks**

Example

In this example, the clock source for the primary clock is BITS-A, and the clock source for the secondary clock is the internal Stratum 3 oscillator.

```
cli:192.168.220.244:root# aux-device backplane-clock-a bits-a
cli:192.168.220.244:root# aux-device backplane-clock-b internal
cli:192.168.220.244:root# show aux-device backplane-clocks
Stratum-3 oscillator          Installed
Backplane clock A source      bitsA
Backplane clock B source      internal
```

Starting and Stopping the HTTP Server

The chassis runs an HTTP server, which allows CudaView users to manage the chassis with their Web browsers. Refer to the *Cuda 12000 IP Access Switch CudaView Administration Guide* for more information on CudaView.

You can start and stop the HTTP server using the **http-server** command. If you stop the HTTP server, the chassis cannot be managed through CudaView. By default, the HTTP server is enabled and running.

To start and stop the HTTP server, perform these tasks:

Task	Command
1. Enter root mode.	**root**
2. Start the HTTP server.	**http-server enable**
3. Stop the HTTP server.	**http-server disable**

Enabling and Disabling Traffic Relay

You can configure processes, such as the HTTP server, to send and receive TCP or UDP packets using an internal address on the Cuda 12000. This method of sending and receiving packets is called *traffic relay*.

If you are running a TFTP server on the Cuda 12000 as part of FastFlow BPM provisioning, you must enable traffic relay for the TFTP server in order to download configuration files to cable modems. The TFTP server sends and receives packets using an internal address. Refer to the FastFlow BPM documentation set for more information on the FastFlow BPM.

The **traffic-relay** command also allows you to configure the Cuda 12000 for in-band management. For example, you can use this command to enable forwarding of Telnet traffic and HTTP traffic using an internal address, thereby allowing you to perform in-band management of the Cuda 12000 using the CLI or CudaView.

To enable or disable traffic relay for a process, perform these tasks:

Task	Command
1. Enter root mode.	**root**
2. Enable traffic relay for a process.	**traffic-relay** {**dns** | **ftp** | **http** | **snmp** | **snmp-trap** | **ssh** | **syslog** | **telnet** | **tftp** | **time_of_day**} [**port** <port>] *Refer to the Cuda 12000 IP Access Switch CLI Reference Guide for details on command syntax.*
3. Disable traffic relay for a process.	**no traffic-relay** {**dns** | **ftp** | **http** | **snmp** | **snmp-trap** | **ssh** | **syslog** | **telnet** | **tftp** | **time_of_day**}
4. Display traffic relay status.	**show traffic-relay**

Note that the **port** argument allows you to specify a port that the specified process uses to listen for incoming requests.

Example

In this example, traffic relay is enabled for Telnet.

```
cli:192.168.208.3:root# traffic-relay telnet
cli:192.168.208.3:root# show traffic-relay

row count: 10

Protocol     State    Port Number
------------ -------- -----------
tftp         enable            69
time_of_day  enable            37
syslog       disable          514
dns          disable           53
snmp         disable          161
telnet       enable            23
ssh          disable           22
http         disable           80
ftp          disable           21
snmp-trap    disable          162

cli:192.168.208.3:root#
```

Broadcasting Messages to Users

The **talk** command enables you to broadcast messages to all chassis users and to enable and disable the ability to broadcast messages.

To broadcast messages to users, perform the following task:

Task	Command
From any mode, send a broadcast message.	**talk** <message-string> *Note that if the string contains spaces, enclose it in quotes.*

To enable or disable the broadcast capability, perform the following task:

Task	Command
From any mode, enable or disable the broadcast capability.	**talk** {**enable	disable**}

Example

```
cli:192.168.208.3:root# talk "Testing the broadcast message feature"

@14:20:42 root@techpubs:: Testing the broadcast message feature

cli:192.168.208.3:root#
```

5 MULTI-CHASSIS SUPPORT

This chapter describes multi-chassis support, and includes the following sections:

About Multi-Chassis Support

The purpose of multi-chassis support is to allow network administrators, from a single session, to access and manage multiple chassis as a single group. When a network administrator connects to a chassis that is a member of a multi-chassis group, the administrator can also access all other chassis in that group without having to connect to each chassis individually.

To become a member of a multi-chassis group, each chassis must have multi-chassis support enabled and must be configured with the same group name. When a chassis is configured with a group name, the chassis registers its name with the Jini™ lookup service, which acts as a directory that enables all chassis in the same group to find each other.

When planning multi-chassis support, consider that each Cuda 12000 can run the Jini lookup service, but the service does not have to be enabled on every Cuda 12000. Refer to "Planning Multi-Chassis Support" on page 96 for more information on planning multi-chassis support.

To access the multi-chassis group, the network administrator logs in to one of the chassis in the group. The network administrator then can select either that chassis to manage or any other chassis in the group.

The Cuda 12000 uses a proxy-based approach to communicate between the host chassis — the chassis to which you are currently logged in — and other chassis within the same group. The Java server on the management module in the host chassis forwards network management messages that are destined for other chassis to the appropriate chassis group member.

You can manage SNMPv3 resources (contexts and users) on the host chassis only. In addition, you can manage hardware alarms/faults and user accounts on the host chassis only. You cannot manage SNMPv3, hardware alarms/faults, and user accounts by proxy.

The multi-chassis group, MCS-Group1, in the example below consists of three chassis: A, B, and C. The group name "MCS-Group1" is configured on each chassis, and each chassis registers this name with the Jini lookup service running on Chassis C. In this example, the network administrator logs in to Chassis B (the host chassis), but keep in mind that the network administrator can also log in to Chassis A and Chassis C to access the group. When the network administrator performs network management operations (such as configuring an interface) on Chassis A and Chassis C, then Chassis B forwards network management messages to Chassis A and Chassis C. Chassis B also forwards messages to the network administrator's workstation that originate from Chassis A and Chassis C.

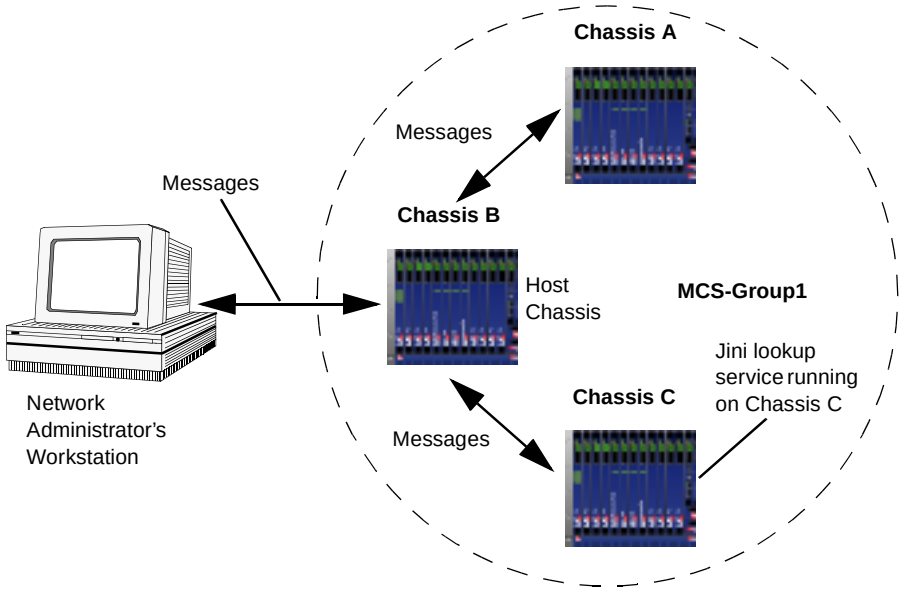


Figure 5-1 Sample Multi-Chassis Group

Planning Multi-Chassis Support

Before you configure multi-chassis support, perform these planning tasks:

- Identify each Cuda 12000 chassis that will be in the group. Make sure that all chassis in the group are running software versions that have multi-chassis support.
- Decide on a group name. A descriptive name is suggested (for example, "MCS-Group-Net-Mgmt"). The name may not contain spaces. You will configure this name on each chassis in the group.
- Decide on a user account for the group with the necessary access privileges for performing your desired network management tasks. All Cuda 12000s in the group must have a common, identical user account. When you connect to a chassis in the group, you must login to this account to access all members of the group.
- Make sure that all chassis in the group share the same physical network (such as the same Ethernet LAN). Members of the same group cannot reside on different physical networks.
- Identify two Cuda 12000 chassis that will run the Jini lookup service. The reason you should enable the Jini lookup service on two Cuda 12000 chassis (instead of one) is to put redundancy in place. Each chassis that runs the Jini lookup service must be on the same physical network as the multi-chassis group that it serves.

After you finish planning multi-chassis support, perform these tasks:

1. Enable the Jini lookup service. Refer to "Enabling the Jini Lookup Service" on page 97.
2. Configure multi-chassis support on each Cuda 12000 chassis in the group. Refer to "Configuring Multi-Chassis Support" on page 98 for details.
3. Create a common user account on each chassis in the group. Refer to "Creating a Common User Account for the Group" on page 100.
4. Monitor the group as needed. Refer to "Viewing Chassis Details" on page 101.

Enabling the Jini Lookup Service

Enable the Jini lookup service on at least two Cuda 12000 chassis. Each chassis that runs the Jini lookup service must be on the same physical network as the multi-chassis group that it serves. To enable the Jini lookup service on a chassis, perform the following tasks:

Tasks	Commands
1. Enter configuration mode.	**root**
2. Enable the Jini lookup service.	**lookup enable**
3. Verify the Jini status.	**show lookup**

Example

In this example, the status of the Jini lookup service is displayed, then the Jini lookup service is started, and finally the Jini lookup status is displayed again.

```
cli:192.168.208.3:root# show lookup
# JINI lookup service (reggie) is stopped.
cli:192.168.208.3:root# lookup enable
Please wait, this may take some time ...
# rmid is stopped
# Starting RMI activation daemon: OK
#
# httpd (pid 4055 4054 4053 4052 4051 4050 4049 540 539 538 537
536 535 534 533
531) is running...
# rmid (pid 5492 5491 5490 5489 5488 5487 5486 5485 5484 5483
5482 5481 5480 547
9 5478 5433) is running...
# JINI lookup service (reggie) is stopped.
# CODEBASE= http://192.168.208.3:80/jini/reggie-dl.jar
# POLICY= /bas/data/java.policy
# LOG_DIR= /var/log/reggie_log
# GROUPS= Cuda12000
# Registering JINI lookup service: OK
#
# JINI lookup service (reggie) is running.
cli:192.168.208.3:root# show lookup
# JINI lookup service (reggie) is running.
```

Configuring Multi-Chassis Support

The Cuda 12000 ships with multi-chassis support enabled. During the initial installation (or upgrade) of the Cuda operating system software, the Java server checks for a multi-chassis support service property. If the property is not found, the Java server automatically enables multi-chassis support, using Jini as the chassis discovery mechanism on the local network.

When multi-chassis support is enabled, all chassis on the same physical network register with each Jini lookup service. Chassis that have the same group name form a multi-chassis group. By default, the Cuda 12000 ships with multi-chassis support activated, and the local group name of Cuda.

You can use the CLI to manually enable and disable multi-chassis support and to override the default group name with one of your own choosing. If you override the default name, make sure that the new name you specify is the correct group name (that is, it matches the group names configured on other chassis in the group). In addition, you can also specify a group description and access additional chassis using the connect command.

Perform these tasks to manually configure multi-service support on a chassis:

Tasks	Commands
1. Enter configuration mode.	**root**
2. Enable or disable multi-chassis support, specify a group name, or specify a description.	**chassis** {**mcs** {**enable** | **disable**} | **group** <group name> | **description** <string>}
3. Display multi-chassis group status.	**show chassis** {**local | <ip-address>**}
4. Access another chassis while in the current log-in session.	**connect** <ip address>

Example

In this example, the administrator enables multi-chassis support, specifies a group name, and displays multi-chassis status on the chassis (local chassis) that the administrator is currently configuring.

```
cli:192.168.208.3:root# chassis mcs enable
cli:192.168.208.3:root# chassis group Cuda-Group1
cli:192.168.208.3:root# chassis description "Cuda Group 1"
cli:192.168.208.3:root# show chassis local

Multi Chassis Service : enable
Host Name   : techpubs
IP Address  : 192.168.208.3
Group Name  : Cuda-Group1
Version     : 3.0.19 Release3.0_Beta 5 2001_09_13_1334
Description : Cuda Group 1
```

Creating a Common User Account for the Group

On each chassis in the group, create the same user account (same username, same password, same access privileges). Then, to access all chassis in the group, log in to the host chassis using this account.

You can manage user accounts on a chassis to which you are directly connected and logged into only (on the host chassis). You cannot manage user accounts by proxy.

Refer to Chapter 3 "Managing User Accounts" for more information on creating user accounts.

Viewing Chassis Details

You can view chassis details for a local chassis, a particular chassis within a group, or all the chassis in a group. Chassis details include the following information:

Table 5-1 Chassis Details

Parameter	Description
Multi Chassis Service	Indicates whether or not multi-chassis support is activated on the particular chassis. This field appears only if you specify the **local** keyword on the **show chassis** command.
Host Name	The host name assigned to the chassis.
IP Address	Indicates the IP address of the particular chassis.
Group Name	The group name assigned to the chassis.
Version	The current Cuda software version that is running on the chassis.
Description	The description assigned to the chassis group.

Perform the following tasks to view chassis details. Note that all commands are issued in **root** mode:

Task	Command
1. View details of the host chassis. The host chassis is the chassis for which you currently have access.	**show chassis local**
2. View details for a particular chassis, within the same group in which you currently have access.	**show chassis** <ip address>
3. View details for all the chassis within the same group in which you currently have access.	**show chassis**

Example

The following is an example of a list of chassis in the group named "Cuda:"

```
cli:192.168.220.208:root# show chassis

Found 33 chassis.

Host Name   : jsl_cuda
IP Address  : xxx.xxx.xxx.xxx
Group Name  : cuda
Version     : 3.0.6 R3dev_cmts 16 2001_07_16_1532
Description : null

Host Name   : lynnebcm
IP Address  : xxx.xxx.xxx.xxx
Group Name  : cuda
Version     : 3.0.6 release3.1_bgp 10 2001_08_10_0647
Description : null

Host Name   : sw19_cuda
IP Address  : xxx.xxx.xxx.xxx
Group Name  : cuda
Version     : 3.0.9 R3dev_cmts 10 2001_07_25_1610
Description : null

Host Name   : adc_cuda
IP Address  : xxx.xxx.xxx.xxx
Group Name  : cuda
Version     : 3.0.9 Release3.1 29 2001_07_24_1130
Description : null

Host Name   : c0222
IP Address  : xxx.xxx.xxx.xxx
Group Name  : cuda
Version     : 3.0.13 Release3.0_Beta 140 2001_08_08_1331
Description : null
```

6 MODULE ADMINISTRATION

This chapter provides general information on how to view and manage Cuda application modules through the CLI and provides specific information on managing Ethernet modules. The chapter includes the following sections:

- Cuda Application Modules (page 104)
- Configuring the 10/100 Ethernet and GigE Modules (page 105)
- Viewing Module Information (page 106)
- Viewing Ethernet Interface Packet Statistics (page 110)

Refer to Chapter 7 "Packet Over SONET Administration" for details on Packet Over SONET administration.

Refer to Chapter 18 "Configuring Cable Modem Termination Systems" for details on CMTS administration.

Cuda Application Modules

Cuda 12000 application modules interface with attached networks. The system supports installation of the following module types:

- **DOCSIS** — Provides Cable Modem Terminating System (CMTS) functions for two-way data communication over domestic cable networks.
- **EuroDOCSIS** — Provides Cable Modem Terminating System (CMTS) functions for two-way data communication over European cable networks.
- **10/100 Octal Ethernet** — Provides eight autosensing 10/100 Mbps. ports for connection to your Ethernet network.
- **Gigabit Ethernet** — Provides 1000 Mbps connection to your Gigabit Ethernet network.
- **Packet-Over-SONET (POS)** — Available in both OC-3 and OC-12 configurations, provides high-speed transmission of IP packets directly over SONET links.

Configuring the 10/100 Ethernet and GigE Modules

The Cuda 12000 allows you to configure duplex mode for interfaces on the 10/100 module and the interface on the GigE modules. The Cuda 12000 also allows you to configure speed for interfaces on the 10/100 module.

You may set duplex mode to full duplex, half duplex, or auto negotiation. You may set the speed of the 10/100 module to 10 Mbps, 100 Mbps, or auto negotiation. By default, the Cuda 12000 sets duplex mode and speed for auto negotiation.

You use the following commands to change duplex mode and speed settings:

Task	Command
1. Enter interface configuration mode for the Ethernet interface.	**interface** <c/s/i>
2. Set interface duplex mode.	**duplex** {**half | full | auto**}
3. Set interface speed.	**speed** {**10 | 100 | auto**}
4. Enable interface auto negotiation.	**negotiation auto**
5. Disable interface auto negotiation.	**no negotiation auto**

Refer to the *Cuda 12000 IP Access Switch CLI Reference Guide* for more information on these commands.

Viewing Module Information

This section provides information on how to view module information, including:

- Viewing Installed Modules
- Viewing Module Versions

Viewing Installed Modules

You can display a listing of modules and their associated interfaces currently installed on the system by using the **show topology** command. To show the current physical configuration of the system, perform the following task from within any mode:

Task	Command
Display system topology, including module and interface information.	**show topology**

You can pipe the output to an include utility to scope the display down to content of interest. For example, the command **show topology | include Ethernet** *will scope the output to only Ethernet interfaces. Note that the string you specify is case-sensitive.*

The following example shows the modules currently installed in the Cuda chassis:

```
cli:172.16.19.10:root# show topology

row count: 12

Chassis/Slot/    Class      Interface Type       Status
Interface
---------------- ---------- -------------------- --------------
1 / 1 / 1        Egress     docsCableMaclayer    Active
1 / 3 / 1        Egress     POS (OC3c)           Not In Service
1 / 6 / 1        Egress     docsCableMaclayer    Active
1 / 8 / 1        Egress     Ethernet (Gigabit)   Not In Service
1 / 11 / 1       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 2       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 3       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 4       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 5       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 6       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 7       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 8       Egress     Ethernet (100 Mb)    Not In Service

cli:172.16.19.10:root#
```

Viewing Module Versions

You can view the software version currently installed on each module. To do so, perform the following task within any mode.

Task	Command
Show the firmware version installed on each module.	**show version**

For example:

```
cli:172.16.19.10:root# show version
2.0.1 Release2 8 2000_10_10_2059

row count: 6

Chassis Slot    LPort   Boot Time      Description
------- ------- ------- -------------- ----------------------------------------
      1       1       2 99-08-28 11:43 BAS CMTS, Hardware V1, Software V2.0,
                                       Build #1 [Release2  8  ] built
                                       2000_10_10_2059
      1       3       1 99-08-28 11:43 BAS Forwarder, Hardware V1, Software
                                       V2.0, Build #1 [Release2  8  ] built
                                       2000_10_10_2059
      1       6       2 99-08-28 11:43 BAS EURO-CMTS, Hardware V1, Software
                                       V2.0, Build #1 [Release2  8  ] built
                                       2000_10_10_2059
      1       8       1 99-08-28 11:43 BAS Forwarder, Hardware V1, Software
                                       V2.0, Build #1 [Release2  8  ] built
                                       2000_10_10_2059
      1      11       1 99-08-28 11:43 BAS Forwarder, Hardware V1, Software
                                       V2.0, Build #1 [Release2  8  ] built
                                       2000_10_10_2059
      1      11       2 99-08-28 11:43 BAS Route Server, Hardware V1, Software
                                       V2.0, Build #1 [Release2  8  ] built
                                       2000_10_10_2059

cli:172.16.19.10:root#
```

Table 6-1 describes the information in the display.

Table 6-1 Show Version Field Descriptions

Field	Description
Chassis	Number assigned to the chassis in which each module resides.
Slot	Number of the physical chassis slot in which the module resides. For information on how slots are numbered, see the *Cuda 12000 IP Access Switch Installation Guide.*
LPort	Logical port utilized by the module. *For ADC use only.*
Boot Time	Indicates date and time of last module bootup.
Description	Specifies the following: ■ Hardware and software version of the module. ■ Module type *(function)* ■ Build Information *(build number and date)*

Viewing Ethernet Interface Packet Statistics

You can view both incoming and outgoing packet statistics for a selected interface. To do so, perform the following tasks within either root mode or interface configuration mode:

Task	Command
1 Show incoming packet statistics for a selected Ethernet interface.	**show interface ethernet** <c/s/i> **in-counters**
2. Show outgoing packet statistics for a selected Ethernet interface.	**show interface ethernet** <c/s/i> **out-counters**

The following example uses the **show topology** command piped to *include* to grep for all Ethernet interfaces. It then displays incoming packet statistics for Ethernet interface 8:

```
cli:172.16.19.10:root# show topology | include Ethernet
1 / 8 / 1        Egress     Ethernet (Gigabit)   Not In Service
1 / 11 / 1       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 2       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 3       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 4       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 5       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 6       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 7       Egress     Ethernet (100 Mb)    Not In Service
1 / 11 / 8       Egress     Ethernet (100 Mb)    Not In Service

cli:172.16.19.10:root# show interface ethernet 1/11/8 in-counters
Interface                          1 / 11 / 8
Type                                 ethernet
In octets                                   0
In unicast                                  0
In multicast                                0
In broadcast                                0

cli:172.16.19.10:root# show interface ethernet 1/11/8 out-counters
Interface                          1 / 11 / 8
Type                                 ethernet
Out octets                                  0
Out unicast                                 0
Out multicast                               0
Out broadcast                               0

cli:172.16.19.10:root#
```

The following incoming statistics are displayed for each interface:

- **In Octets** — Total number of Octets that have been received on this interface, including framing characters.
- **In Unicast Packets** — Number of Unicast packets that have been received on this interface.
- **In Multicast Packets** — Number of Multicast packets that have been received on this interface.
- **In Broadcast Packets** — Number of Broadcast packets that have been received on this interface.

The following outgoing statistics are displayed for each interface:

- **Out Octets** — The total number of octets that have been transmitted out of this interface, including framing characters.
- **Out Unicast Packets** — The total number of Unicast packets that have been transmitted out of this interface.

- **Out Multicast Packets** — The total number of Multicast packets that have been transmitted out of this interface.
- **Out Broadcast Packets** — The total number of Broadcast packets that have been transmitted out of this interface.

Displaying Statistics for All System Interfaces

You can display incoming and outgoing statistics for all system interfaces. To do so, perform the following tasks within any mode:

Task	Command
1 Show incoming packet statistics for a selected Ethernet interface.	**show in-counters**
2. Show outgoing packet statistics for a selected Ethernet interface.	**show out-counters**

The following example displays *incoming* statistics for all system interfaces, then pipes the output to include *outgoing* statistics for only cable (DOCSIS) interfaces:

```
cli:172.16.19.10:root# show in-counters

row count: 22

Interface     Type           In octets  In unicast In multicast In broadcast
------------ -------------- ---------- ---------- ------------ ------------
    1 / 1 / 1   docsCableMac   11512319     322192            0            0
    1 / 1 / 3 docsCableUS(1)    6613451     186371            0            0
    1 / 1 / 4 docsCableUS(2)    4898868     135821            0            0
    1 / 1 / 5 docsCableUS(3)          0          0            0            0
    1 / 1 / 6 docsCableUS(4)          0          0            0            0
    1 / 3 / 1          sonet          0          0            0            0
    1 / 3 / 2      sonetPath          0          0            0            0
    1 / 3 / 3            ppp          0          0            0            0
    1 / 6 / 1   docsCableMac          0          0            0            0
    1 / 6 / 3 docsCableUS(1)          0          0            0            0
    1 / 6 / 4 docsCableUS(2)          0          0            0            0
    1 / 6 / 5 docsCableUS(3)          0          0            0            0
    1 / 6 / 6 docsCableUS(4)          0          0            0            0
    1 / 8 / 1       ethernet          0          0            0            0
   1 / 11 / 1       ethernet          0          0            0            0
   1 / 11 / 2       ethernet          0          0            0            0
   1 / 11 / 3       ethernet          0          0            0            0
   1 / 11 / 4       ethernet          0          0            0            0
   1 / 11 / 5       ethernet          0          0            0            0
   1 / 11 / 6       ethernet          0          0            0            0
   1 / 11 / 7       ethernet          0          0            0            0
   1 / 11 / 8       ethernet          0          0            0            0

cli:172.16.19.10:root# show out-counters | include docs
    1 / 1 / 1   docsCableMac  1808872424     315544     330207364         7669
    1 / 1 / 2    docsCableDS  1808873040     315544     330207378         7669
    1 / 6 / 1   docsCableMac           0          0             0            0
    1 / 6 / 2    docsCableDS           0          0             0            0

cli:172.16.19.10:root#
```

7 PACKET OVER SONET ADMINISTRATION

This chapter provides information on how to configure Packet over SONET (POS) on the Cuda 12000 using the CLI and includes the following sections:

The section covers functionality available in the Cuda 12000 Base System Software.

About Packet Over SONET

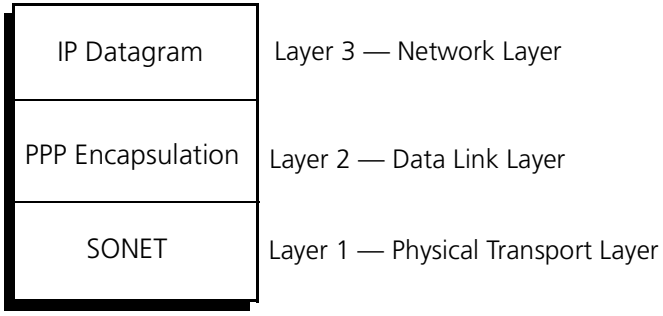
Packet Over SONET enables the Cuda 12000 to transmit IP packets over SONET links; essentially placing the IP layer over the SONET physical layer. POS makes efficient use of bandwidth, allowing for lower packet overhead and extremely fast transmission speeds.

The system uses point-to-point protocol (PPP) to transport IP data over SONET point-to-point circuits, as described in RFC 2615. The IP over SONET transmission process consists of three primary steps:

- Encapsulate the IP datagram into a PPP frame.
- Place the PPP frame into the payload portion of the SONET frame.
- Transmit the SONET frame over the point-to-point circuit.

Figure 6-1 shows the POS transport structure in relation to the OSI network model:

Figure 6-1 Packet Over SONET — Network Structure



POS administration on the Cuda 12000 involves the following:

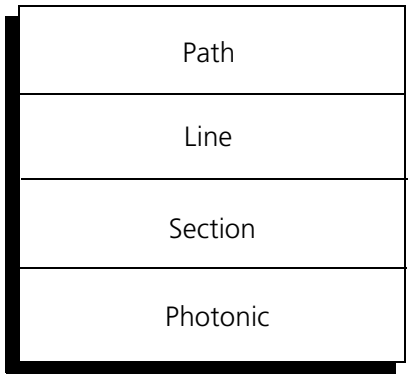
- Administration of the physical SONET interface at layer 1, as described in "Packet Over SONET (POS) Interface Administration," next.
- Administration of the point-to-point protocol (PPP) used to encapsulate the IP data at layer 2, as described in "Configuring Point-to-Point Protocol (PPP)" on page 137.

Packet Over SONET (POS) Interface Administration

Packet over Synchronous Optical Network (SONET) allows for high-speed transport of IP data packets over a SONET network. The OC-3 and OC-12 POS modules contain a single physical interface that supports connection to STS networks and supports transmission speeds of up to 155 Mbps.

A SONET frame is 810 bytes represented as a grid of 9 rows by 90 columns. The frame consists of hierarchal layers, each providing services for the layer above it. Figure 6-2 shows the logical representation of the SONET layers.

Figure 6-2 SONET Network Structure



The layers that comprise a SONET frame include:

- **Path Layer** — Maps the payload into the synchronous payload envelope (SPE) of the SONET frame and creates the STS-1 synchronous payload envelope (SPE). In POS transmission, the payload contained in the SPE is the PPP encapsulated IP datagram. It then passes the resulting STS-1 SPE to the Line layer.
- **Line Layer** — Combines 3 STS-1 SPEs and adds the appropriate line overhead. This multiplexing of 3 STS-1 SPEs is also referred to as concatenation. It then passes the concatenated SPE to the section layer.
- **Section Layer** — Adds section overhead, performs scrambling, and creates the actual STS frame, which it then passes to the photonic layer.

- **Photonic Layer** — Converts the electrical STS-n signal to an optical signal, referred to as Optical Carrier. This OC-n signal is then transmitted over the circuit.

Each layer consists of its own overhead bytes. This overhead provides the powerful management and fault-tolerance capabilities inherent in a SONET network.

SONET overhead also provides for various alarms and error messages — known as defects — to be reported. Alarms allow for the reporting of network failures; error messages report incomplete failures that may compromise data transmission.

SONET interface administration on the Cuda 12000 includes:

- Displaying POS Interface Information
- Disabling and Enabling Interfaces
- Viewing POS Interface Packet Statistics
- Viewing SONET Line-Layer Information
- Viewing SONET Path Layer Information
- Viewing and Configuration Section Layer Administration
- Configuring and Viewing SONET Alarms

Displaying POS Interface Information

You can display information for each POS interface. To do so, perform the following task within root mode:

Task	Command
Display POS interface statistics and settings.	**show interface pos** <c/s/i>

The following example uses the **show topology** command to obtain a list of modules installed on the system, then uses the **show interface** command to display information for the select POS interface.

```
cli:192.168.208.3:root# show topology

row count: 11
Chassis/Slot/    Class      Interface Type       Status
Interface
---------------- ---------- -------------------- --------------
1 / 1 / 1        Egress     docsCableMaclayer    Active
1 / 3 / 1        Egress     Ethernet (Gigabit)   Not In Service
1 / 8 / 1        Egress     POS (OC3c)           Not In Service

cli:192.168.208.3:root# show interface pos 1/8/1
-------------------------------------------------------
Interface Type                          sonet
POS 1/8/1 (line protocol)              closed
-------------------------------------------------------
Hardware is Packet Over Sonet
Internet Address                  155.144.1.1
Rx Giants                                0
Bad FCS's                                0
Bad Addresses                            0
Bad Controls                             0
Local MRU                             1500  (bytes)
Remote MRU                            1500  (bytes)
FCS Size                                32  (bits)
Transmission Errors (Tx)                 0
Rx Abort                                 0
Rx Runts                                 0
Interface Type                         ppp
Interface Speed                     155520  (Kbits)
Interface In Octets                      0
Interface In Unicast Pkts                0
Interface In Multicast Pkt               0
```

```
Interface In Broadcast Pkt                0
Interface In Discards                     0
Interface In Errors                       0
Interface Out Octets                      0
Interface Out Unicast Pkts                0
Interface Out Multicast Pk                0
Interface Out Broadcast Pk                0
Interface Out Discards                    0
Interface Out Errors                      0
LCP                                  closed
Open                                   None
Negotiation Attempts                     10
Retry Timeout                             3
IPCP IP Address Report                Enabled
Framing                                 Sonet
Line Type                  Single Mode (15km)
Clock Source                            Line
Path Signal Id - C2                     0xCF
Section Trace Byte - J0                 0xCC
Packet Scrambling                   Disabled
Loopback                   None
Last clearing of counters           431:47:9  (HH:MM:SS)
PPP Authentication
  Security Mode: NONE
```

The display includes a number of statistics, as described in the following table.

Table 7-1 POS Interface Statistics

Display Element	Description
Interface Type	Number representing Packet over SONET.
POS 1/3/1 (line protocol)	Indicates whether a SONET line-layer connection is open or closed.
Hardware is Packet over SONET	Indicates hardware module type.
Internet Address	IP address of the POS interface.
Rx Giants	The number of packets received on this link which are larger than the maximum packet size.
Bad FCS's	This LCP statistic indicates the number of received packets that have been discarded due to having an incorrect FCS.
Bad Addresses	Number of packets discarded because they were received with incorrect address or control fields. Address field was not 0xFF or control field was not 0x03.
Bad Controls	Number of packets received on this link with an incorrect Control Field.
Local MRU	Local Maximum Receive Unit (MRU), which is the current value of the MRU for the local PPP Entity. The remote entity uses this MRU when sending packets to the local PPP entity. Value is only meaningful only when the link has reached the *open* state
FCS Size	Frame Check Sequence (FCS) in bits that the local entity generate s when sending and receiving packets to and from the remote entity. Value is only meaningful when the link has reached the open state.
Transmission Errors (Tx)	This Line-layer statistic calculates the sum of all transmit errors that caused the packet to not be transmitted. These errors consist of tx fifo error, link layer errors, minimum packet size violations, maximum packet size violations and tx parity errors.

Display Element	Description
Rx Abort	The number of packets received on this link in which the abort sequence is detected.
Rx Runts	The number of packets received on this link which are smaller than the minimum packet size.
Interface Type	Displays "ppp" (Point-to-Point Protocol).
Interface Speed	Transmission speed in Kbits.
Interface in Unicast Pkts	Number of Unicast packets received on this interface.
Interface in Discards	Total number of incoming packets that were discarded by this interface.
Interface in Errors	Number of incoming packet errors seen on this interface.
Interface Out Unicast Pkts	Number of Unicast packets transmitted from this interface.
Interface Out Discards	Total number of outgoing packets that were discarded by this interface.
Interface Out Errors	Number of outgoing packet errors seen on this interface.
LCP	Indicates whether the LCP layer is open or closed on this interface.
Open	Indicates which network layer protocols are open. For example IPCP would be displayed when IP is up.
Negotiation Attempts	The maximum number of link negotiation attempts allowed by this interface.
Retry Timeout	Interval to wait between link connection retries.
IPCP IP Address Report	Indicates whether NCP is enabled for the transmission of IP datagrams. IP Control Protocol (IPCP) is the network control protocol used.
Line Type	Indicates SONET line type.
Clock Source	Indicates the clock source configured for this interface.
Path Signal Id - C2 byte	C2 byte received in last packet.
Packet Scrambling	Indicates whether packet scrambling in enabled or disabled.

Display Element	Description
Loopback	Indicates loopback configuration.
Last clearing of counters	Time *(SysUpTime)* since the counters were last cleared and reset to zero.
PPP Authentication Security Mode	Indicates authentication used on this interface — PAP or CHAP

Clearing Interface Counters

You can clear interface counters for a selected POS interface. To do so, perform the following task within interface pos <c/s/i> mode:

Task	Command
Clear all counters for the current POS interface.	clear counters

Disabling and Enabling Interfaces

You can manually take an interface offline or bring it online. When an interface is enabled *(online)*, it can forward traffic; when disabled *(offline)* it cannot.

***NOTE**: Enabling a POS interface assumes the POS interface is already configured.*

To disable and enable a POS interface, perform the following tasks in interface pos <c/s/i> mode:

Task	Command
1. Disable the POS interface so that the administrative status is *down*.	**shutdown**
2. Enable the POS interface so that the administrative status is *up*.	**no shutdown**

In this example, the user disables the POS interface in slot 3 and then brings it back online:

```
cli:172.16.19.10:root# interface 1/3/1
mode: interface:pos:csi(1/3/1)
cli:172.16.19.10:interface:pos:csi(1/3/1)# shutdown
cli:172.16.19.10:interface:pos:csi(1/3/1)# no shutdown
```

Viewing POS Interface Packet Statistics

You can view incoming and outgoing packet statistics for selected POS interfaces. These traffic statistics provide a snapshot overview as to the amount and type of traffic flowing across the interface.

For each POS interface, incoming and outgoing statistics are shown for the physical SONET layer, the Path layer, and at the PPP layer. To display these statistics, perform the following tasks:

Task	Command
1. From within root mode, issue this command to display incoming statistics:	**show interface pos** <c/s/i> **in-counters**
2. From within interface mode, issue this command to display incoming statistics:	**show in-counters**
3. From within root mode, issue this command to display outgoing statistics:	**show interface pos** <c/s/i> **out-counters**
4. From within root mode, issue this command to display outgoing statistics:	**show out-counters**

The following incoming statistics are displayed for the SONET, Path, and PPP layer on each POS interface:

- **In Octets** — Total number of PPP negotiations octets that have been received on this interface. *This does not include octets for data packets.*
- **In Unicast Packets** — Number of Unicast packets that have been received on this interface.
- **In Multicast Packets** — Number of Multicast packets that have been received on this interface.
- **In Broadcast Packets** — Number of Broadcast packets that have been received on this interface.

The following outgoing statistics are displayed for each interface:

- **Out Octets** — Total number of PPP negotiations octets that have been transmitted from this interface. *This does not include octets for data packets.*
- **Out Unicast Packets** — Total number of Unicast packets that have been transmitted from this interface.
- **Out Multicast Packets** — Total number of Multicast packets that have been transmitted from this interface.
- **Out Broadcast Packets** — Total number of Broadcast packets that have been transmitted from this interface.

Viewing SONET Line-Layer Information

The SONET Line layer serves as the path between multiplexers and is responsible for synchronizing data transmission and multiplexing the STS-*n* signals generated by the section layer.

Performance management statistics are collected at the SONET line layer. To view these Line-layer statistics, perform the following task within root mode:

Task	Command
View SONET Line information.	**show controllers pos** <c/s/i> **| include Line**

The **show controllers pos** command includes a variety of information. The following example displays Line-layer statistics by piping the output to include only *Line* statistics:

```
cli:172.16.19.10:root# show controllers pos 1/3/1 | include Line
Line: AIS                                       0
Line: RDI                                       0
Line: FEBE(M1)                                  0
Line: BIP(B2)                                   0

cli:172.16.19.10:root#
```

To restart the statistics counters to zero, issue the **clear counters** command within POS interface configuration mode.

Table 7-2 SONET Line Layer Statistics

Display Element	Description
AIS — Path Alarm Indication Signal Detections (PAIS)	The number of times a Path Alarm Indication Signal has been detected. A PAIS occurs if all the H1/H2 pointer bytes in the received SONET frame are 01.
RDI — Path Remote Defect Indication Detections (PRDI)	The number of times a Path Remote Defect Indication has been detected. A PRDI occurs if bits 5,6 and 7 of the G1 byte received with the same value for 5 consecutive frames.
FEBE (M1)	Far end block errors - Indicates the number of B2 errors that were detected by the remote side in its receive signal.

Display Element	Description
BIP (B2)	Even Parity is calculated over groups of 3 bytes of each frame, except the first 3 rows of TOH. The value is compared to the B2 values in the received frame. Mismatches are counted.

Viewing SONET Path Layer Information

The Path layer is responsible for mapping the data to be transported into the synchronous payload envelope (SPE) of the SONET frame. It creates the STS-1 SPE and passes it to the line layer.

You can view Path-layer performance information for a selected POS interface. This information includes defects and error statistics to provide an assessment of Path layer operation. To view these Path-layer statistics, perform the following task within root mode:

Task	Command
View SONET Path layer information.	**show controllers pos** <c/s/i> **| include Path**

The `show controllers pos` command includes a variety of information. The following example displays Path-layer statistics by piping the output to include only *Path* statistics:

```
cli:172.16.19.10:root# show controllers pos 1/3/1 | include Path
Path: AIS                                    0
Path: RDI                                    0
Path: FEBE(G1)                               0
Path: BIP(B3)                                0
Path: LOP                                    0

cli:172.16.19.10:root#
```

To restart the statistics counters to zero, issue the **clear counters** command within POS interface configuration mode.

Table 7-3 describes the SONET Path statistics shown in the display.

Table 7-3 SONET Path Layer Statistics

Display Element	Description
AIS — Path Alarm Indication Signal Detections (PAIS)	The number of times a Path Alarm Indication Signal has been detected. A PAIS occurs if all the H1/H2 pointer bytes in the received SONET frame are 01.
RDI — Path Remote Defect Indication Detections (PRDI)	The number of times a Path Remote Defect Indication has been detected. A PRDI occurs if bits 5,6 and 7 of the G1 byte received with the same value for 5 consecutive frames.
FEBE (G1)	Far end block errors - Indicates the number of B3 errors that were detected by the remote side in its receive signal.
BIP (B3)	Even Parity is calculated over all bits in the SPE, including POH of each frame. These values are then compared to the B3 values received in the packet. Mismatches are reported.

Section Layer Administration

The primary roles of the section layer include synchronization and timing of the SONET transmission, and passing the electrical STS-*n* frame format to the photonic layer where it is then converted to an optical signal and transported to the adjacent device.

Section layer administration involves viewing the current status of the configuration and modifying the configuration of the section layer parameters for a selected POS interface. You can configure the following POS section layer parameters:

Loopback Configuration

Loopback configuration on a POS interface allows you to test interface connectivity and connection to a remote device. By default, loopback is not configured. The system supports the following loopback configuration:

- **Line** — Configures the POS interface to loop-back data to the originating device. While configured in this mode, the interface loops back and retransmits incoming data without actually receiving it.
- **Internal** — Configures the POS interface to loop-back data to itself. While configured in this mode, the interface loops-back outgoing data to the receiver without actually transmitting it.

To configure loopback testing on a selected POS interface, perform the following tasks in interface pos <c/s/i> mode:

Task	Command
1. Enable loopback testing.	**loop** {**line** | **internal**}
2. Disable loopback testing.	**no loop**

Clock Source

SONET is a synchronous transport technology. Timing for this synchronous transmission of data is derived from one of the following clock sources:

- **Line** — Also referred to as loop timing, this timing option configures the interface to use the recovered receive clock to provide transmit clocking. This is the default clock source.
- **Internal** — Configures the interface to generate the transmit clock internally.

Keep in mind that the clock source you configure for the POS interface can also be used as the clock source for the primary or secondary backplane clock. If you want to use the clock source for the POS interface as the clock source for the primary or secondary backplane clock, you must configure the Line clock source for the POS interface. Refer to "Configuring Clock Sources" on page 86 for more information on backplane clocks.

To configure the clock source for a selected POS interface, perform the following task in interface pos <c/s/i> mode:

Task	Command
Configure the clock source for the current interface.	**clock-source** {**line** | **internal**}

When configuring point to point links, one side of the link should be configured to utilize a line clock source, the other should utilize an internal clock source.

Signal Type

Configures the type of signal *(framing)* this POS interface transmits. Currently, the system supports only SONET STS-*n* framing.

To configure the framing type used on a POS interface, perform the following task in interface pos <c/s/i> mode:

Task	Command
Configure POS framing.	**pos framing** {**sonet** | **sdh**}

Packet Scrambling

Enables scrambling of SONET Synchronous Payload Envelopes (SPEs) on this interface. Note that both end-points of the transmission must use the same scrambling. Scrambling is disabled by default.

To configure scrambling on a POS interface, perform the following task in interface pos <c/s/i> mode:

Task	Command
Enable scrambling on the POS interface.	**pos scramble**

Note the following:

- *Only normal* ***Path Remote Defect Indication*** *is currently supported on POS interfaces.*
- *Only* ***SONET*** *signal (framing) is supported on POS interfaces.* Configuring and Viewing SONET Alarms

Configuring and Viewing SONET Alarms

A major advantage of SONET is that it can generate alarm and error messages when problems occur, such as when a signal fails or degrades.

A receiving interface is notified of network defects in the form of Alarm Indication Signals (AIS); transmitting interfaces are notified of network defects by the return of Remote Defect Indications (RDI).

You can configure the alarms and defects that you want the selected POS interface to report. SONET alarm administration on a POS interface involves the following:

- **Configuring POS Alarm Reporting** — You configure the Alarms that you want the POS interface to report on using the **pos report** command within cable interface configuration mode.
- **Viewing Alarm Information** — You can verify the alarms that are enabled and disabled on a selected POS interface, as well as the alarms that have been reported, using the **show controller pos** <c/s/i> command within root mode.

Configuring POS Alarm Reporting

You can configure reporting of 12 different POS alarms. To do so, perform the following tasks within interface pos <c/s/i> mode:

Alarm Report	Description	Command
Line Alarm Indication Signal (LAIS)	Disabled by default, configures the interface to report line alarm indication signal errors.	■ To enable reporting: **pos report lais** ■ To disable reporting: **no pos report lais**
Line Remote Defect Indication (LRDI)	Disabled by default, configures the interface to report line remote defect indication errors.	■ To enable reporting: **pos report lrdi** ■ To disable reporting: **no pos report lrdi**
Path Alarm Indication Signal (PAIS)	Disabled by default, configures the system to report path alarm indication signal errors. Line terminating equipment (LTE) send packet alarm indication signals to alert downstream path terminating equipment (PTE) of defects on their incoming line signal.	■ To enable reporting: **pos report pais** ■ To disable reporting: **no pos report pais**
Path Loss of Pointer (PLOP)	Enabled by default, configures the interface to report path loss of pointer errors. A PLOP error may result from an invalid pointer or too many new data flag enabled indications.	■ To enable reporting: **pos report plop** ■ To disable reporting: **no pos report plop**
Path Remote Defect Indication (PRDI)	Disabled by default, configures the interface to report path remote defect indication errors.	■ To enable reporting: **pos report prdi** ■ To disable reporting: **no pos report prdi**
B2 Signal Degrade (SD)	Disabled by default, configures the interface to report when the B2 signal degrades enough to meet or cross a specified Bit Error Rate (BER) threshold. The default BER threshold for B2 signal failure is 10-6.	■ To enable reporting: **pos report sd-ber** ■ To disable reporting: **no pos report sd-ber** ■ To set the B2 Signal Degrade threshold: **pos threshold sd-ber** <number>

Alarm Report	Description	Command
B2 Signal Fail (SF)	Enabled by default, configures the interface to report a failure when the B2 signal degrades enough to meet or cross a specified Bit Error Rate (BER) threshold. The default BER threshold for B2 signal failure is 10-3.	■ To enable reporting: **pos report sf-ber** ■ To disable reporting: **no pos report sf-ber** ■ To set the B2 Signal Fail threshold: **pos threshold sf-ber** <number>
Loss of Frame (SLOF)	Enabled by default, configures the interface to report section loss of frame errors. The interface detects SLOF when a severely error framing defect on the incoming SONET signal persists for at least 3 milliseconds.	■ To enable reporting: **pos report slof** ■ To disable reporting: **no pos report slof**
Loss of Signal (SLOS)	Enabled by default, configures the interface to report loss of signal (SLOS) errors. The POS interface reports a SLOS error under either of the following conditions: ■ When an all-zeros pattern on the incoming SONET signal lasts at least 19(+-3) microseconds. ■ If the signal level drops below the a specified threshold.	■ To enable reporting: **pos report slos** ■ To disable reporting: **no pos report slos**

Viewing Alarm Information

Using the **show controllers pos** <c/s/i> command within root mode, you can display both the alarms that you have enabled on the POS interface, and whether or not specific alarms have been reported.

To view the alarm reporting configuration on a POS interface, perform the following task in root mode:

Task	Command
View whether the reporting of each POS alarm is enabled or disabled.	**show controllers pos** <c/s/i> | **include alarm**

The following example views the alarm reporting configuration for POS interface 1/3/1:

```
cli:172.16.19.10:root# show controllers pos 1/3/1 | include alarm
Report alarms for B1                  enabled
Report alarms for B2                  enabled
Report alarms for B3                  enabled
Report alarms for LAIS               disabled
Report alarms for LRDI               disabled
Report alarms for PAIS               disabled
Report alarms for PLOP                enabled
Report alarms for PRDI               disabled
Report alarms for SD-BER             disabled
Report alarms for SF-BER              enabled
Report alarms for SLOF                enabled
Report alarms for SLOS                enabled
Local alarm active now                   None
Remote alarm active now                  None

cli:172.16.19.10:root#
```

To view the alarm reporting configuration on a POS interface, perform the following task in root mode:

Task	Command
View whether the reporting of each POS alarm is *enabled* or *disabled*.	**show controllers pos** <c/s/i> | **include alarm**

The following example indicates whether or not a specific alarm has been reported:

```
cli:172.16.19.10:root# show controllers pos 1/3/1 | include now
Local alarm active now                  None
Remote alarm active now                 None
B1 errors occurring now                false
B2 errors occurring now                false
B3 errors occurring now                false
LAIS errors occurring now              false
LRDI errors occurring now              false
PAIS errors occurring now              false
PLOP errors occurring now              false
PRDI errors occurring now              false
SD-BER errs occurring now              false
SF-BER errs occurring now              false
SLOF errors occurring now              false
SLOS errors occurring now              false

cli:172.16.19.10:root#
```

Configuring Point-to-Point Protocol (PPP)

PPP is well-suited for delivery of data over SONET networks, as SONET links are provisioned as point-to-point circuits. The system encapsulates IP datagrams using PPP, then places the PPP frames into the SONET payload before transmission over the SONET circuit. PPP also provides security protocols that support the authentication of peers.

PPP administration on a POS interface includes:

- **Configuring PPP Security** — POS interfaces support both Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) so that only trusted devices can participate in the creation of a point-to-point circuit.
- **Configuring LCP** — As part of establishing the PPP connection, a POS interface uses Link Control Protocol (LCP) packets to configure and test the data link.
- **Enabling NCP** — A Network Control Protocol (NCP) is used to configure and enable network layer protocol communication. In this case, the network layer protocol used over the SONET circuit is IP; the NCP used to enable transmission of IP datagrams is the IP Control Protocol (IPCP).

PPP encapsulation over SONET is described in RFC 2615.

Configuring PPP Security

Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) provide authentication mechanisms that serve to identify the peers that want to establish point-to-point connections. Using both CHAP and PAP, the device must provide a known *username* and *password* to the POS interface with which it wants to establish a PPP connection.

CHAP is more secure than PAP. CHAP clients respond to challenges with an encrypted version of the password; PAP sends unencrypted straight text over the network. In addition, CHAP calls for both endpoints to perform a computation to arrive at a secret string; PAP does not. You can configure the POS interface to attempt authentication using one protocol, and if refused, attempt authorization with the other.

SONET connections are provisioned as point-to-point circuits. The connection is initiated by one peer — the caller — into an adjacent peer — the callee. The caller is referred to as the client; and the callee is referred to as the server. Each CHAP and PAP must be enabled at both endpoints of a point-to-point connection and configured to operate in both client and server mode, as described in the following sections.

Both CHAP and PAP are specified in RFC 1334.

Configuring Client-Side Security Parameters

When initiating a point-to-point connection, the POS interface acts as a client and calls into a remote end-point, which functions as a PPP server. If PAP, CHAP, or both forms of authentication are enabled on the server, then the same authentication protocols must be enabled on the POS interface.

The POS interface, acting as a client, must provide the remote server with the correct username and password. If the interface fails to provide the correct information, the remote device will not allow it to call in and establish a connection.

You can enable CHAP and PAP client-side authentication and configure the security information — *username and password* — that the POS interface sends to a PPP server when initiating a point-to-point connection.

To configure CHAP authentication, perform the following tasks within interface pos <c/s/i> mode:

Task	Command
1. Enable CHAP authentication.	■ To enable the use of CHAP only: **ppp authentication chap** *The interface will use CHAP authentication only; no negotiation.* ■ To enable CHAP then PAP: **ppp authentication chap pap** *The interface will negotiate the authentication protocol to use. It will try to agree on CHAP authentication first, then PAP second.* ■ To enable PAP then CHAP: **ppp authentication pap chap** *The interface will negotiate the authentication protocol to use. It will try to agree on PAP authentication first, then CHAP second.*
2. Configure the hostname the POS interface will use to respond to CHAP Challenges.	**ppp chap-hostname** <name>
3. Configure the password the POS interface will use to respond to CHAP Challenges.	**ppp chap-password** <password>

To configure PAP authentication, perform the following tasks within interface pos <c/s/i> mode:

Task	Command
1. Enable PAP authentication.	■ To enable the use of PAP only: **ppp authentication pap** *The interface will use PAP authentication only; no negotiation.* ■ To enable PAP then CHAP: **ppp authentication pap chap** *The interface will negotiate the authentication protocol to use. It will try to agree on PAP authentication first, then CHAP second.* ■ To enable CHAP then PAP: **ppp authentication chap pap** *The interface will negotiate the authentication protocol to use. It will try to agree on CHAP authentication first, then PAP second.*
2. Configure the username and password that the POS interface will use to respond to PAP Challenges.	**ppp pap-sent-username** <name> **password** <password>

Note that you can disable authentication on a selected POS interface by using the following command within interface pos <c/s/i> mode:

no ppp authentication

Configuring Server-Side Security Parameters

When a remote peer *(client)* calls into the POS interface and attempts to establish a point-to-point connection, the interface functions as a PPP access server. Enabling server-side authentication configures the POS interface to authenticate all peers that call into it.

Configuring server-side authentication involves the following:

- Specifying which protocol you want the interface to use to authenticate clients. You can configure the interface to request CHAP authentication, PAP authentication, or both in a specified order.
- Specifying the hostname the POS interface sends to a client when performing CHAP authentication.
- Adding users to the *PPP Server Users Table.* User account information includes a username and password. When a remote client responds to a PAP challenge with a username and password, the system examines this table to verify that the client has responded with the correct information. If so, the connection is allowed; otherwise, the connection is closed.

Perform the following tasks within interface pos <c/s/i> mode to configure PPP CHAP server-side security parameters:

Task	Command
1. Enable CHAP authentication.	■ To enable the use of CHAP only: **ppp authentication chap** *The interface will use CHAP authentication only; no negotiation.* ■ To enable CHAP then PAP: **ppp authentication chap pap** *The interface will negotiate the authentication protocol to use. It will try to agree on CHAP authentication first, then PAP second.* ■ To enable PAP then CHAP: **ppp authentication pap chap** *The interface will negotiate the authentication protocol to use. It will try to agree on PAP authentication first, then CHAP second.*
2. If you've enable CHAP authentication, configure the hostname that the interface sends to a client.	**ppp chap-hostname** <name>
3. If you've enabled CHAP, configure the username and password that a requesting client must provide for authentication to allow the connection.	**ppp username** <name> **password** <password>

To configure the interface to authenticate peers using PAP, then you must add user account information for all peers that the interface may authenticate.

When PAP is enabled for server mode, the interface requests a username and password from the remote peer. When the peer responds with a username/password combination, the POS interface examines its *PPP LCP Server Users Table* to verify the information is correct. If the account information is verified correct, the connection is allowed; otherwise it's closed.

Perform the following tasks within interface pos <c/s/i> mode to configure PPP PAP server-side security parameters:

Task	Command
1. Enable PAP authentication	■ To enable the use of PAP only: **ppp authentication pap** *The interface will use PAP authentication only; no negotiation.* ■ To enable PAP then CHAP: **ppp authentication pap chap** *The interface will negotiate the authentication protocol to use. It will try to agree on PAP authentication first, then CHAP second.* ■ To enable CHAP then PAP: **ppp authentication chap pap** *The interface will negotiate the authentication protocol to use. It will try to agree on CHAP authentication first, then PAP second.*
2. Configure the username and password that a requesting client must provide for authentication to allow the connection.	**ppp username** <name> **password** <password>

Configuring LCP

The PPP protocol suite includes a Link Control Protocol (LCP) for establishing, configuring and verifying point-to-point connections. PPP uses LCP to determine encapsulation options, set limits in transmit and receive packet size, detect link configuration errors, and terminate links.

LCP is defined in RFCs 1570 and 1661.

Configuring LCP Parameters

To configure LCP parameters for a selected PPP interface, perform the following tasks within interface pos <c/s/i> mode:

Initial Maximum Transmit / Receive Unit (MTU)

Because IP packets are encapsulated in PPP, the maximum length of an IP packet that can be transmitted over the PPP link is the same length as the PPP information field. If a packet is larger than the PPP information field, it must be fragmented a placed in multiple PPP packets. Perform the following task within interface pos <c/s/i> mode to enter the maximum transmit and receive packet size allowed on this interface.

NOTE: *The only MTU size the Cuda 12000 currently supports is 1500. This value should not be changed.*

Task	Command
Configure the maximum transmission unit.	**mtu 1500**

Frame Check Sequence (FCS) Size

Perform the following task within interface pos <c/s/i> `to` enter the frame check sequence size:

Task	Command
Configure frame check sequence size.	**crc** {**16** | **32**}

Max Negotiation Attempts

Perform the following task within interface pos <c/s/i> mode to configure the maximum number of link negotiation attempts allowed by the current interface:

Task	Command
Configure maximum negotiation attempts.	**ppp negotiation-count** <0...100>

Time Between Negotiation Attempts

Perform the following task within interface pos <c/s/i> mode to configure the number of seconds that the interface waits between LCP negotiations.

Task	Command
Configure the time (seconds) between negotiation attempts.	**timeout** <0 ... 65535>

Enabling NCP

IP Control Protocol (IPCP) is the Network Control Protocol (NCP) used to configure, enable, and disable IP protocol access on both ends of a SONET point-to-point circuit. In order for IP packets to be transmitted over the point-to-point link, IPCP must reach the open state. This enables IP communication between the two circuit endpoints.

By default, the Cuda 12000 is configured to provide its IP address during IPCP negotiations. But when negotiating with a Juniper Networks system, providing the IP address during IPCP negotiation prevents a successful connection.

When the interface must connect with a Juniper Networks system, you can disable reporting of an IP address during IPCP negotiation.

To enable or disable reporting of the IP address during negotiation, perform the following task within interface pos <c/s/i> mode:

Task	Command
1. Enable the reporting of the IP address during IPCP negotiation.	**ppp ipcp-report-address**
2. Disable the reporting of the IP address during IPCP negotiation.	**no ppp ipcp-report-address**

8 TIMING AND ALARM CONTROLLER MANAGEMENT

The Cuda 12000 utilizes an external fan tray for cooling and obtains power from an external power source. Fault management features on the Cuda 12000 for the fan tray and power source are:

- The Timing and Alarms Controller (TAC) that resides on the Management module. TAC provides alarm processing for detection of faults associated with fan tray and power supply auxiliary devices.
- Software reporting functions that allow you to configure the reporting of fault conditions. Fault reporting alerts you to fault conditions as they arise, so you may take action prior to a loss of operation, or know when the power source and cooling capability is compromised.
- DB-15 connectors on the rear panel of the Cuda 12000 that allow you to send specific types of alarm signals to notify an auxiliary device that a fault occurred.

This chapter provides information and procedures for configuring the monitoring and reporting of power and fan tray fault conditions and includes the following sections:

About Timing and Alarm Controller Fault Reporting

For a single chassis, you can connect the following units:

- **Fan Tray** — The fan tray serves as the system cooling unit. This is a required component and ships with every Cuda 12000.
- **Power Supply A** — A single -48 volt DC power source is required for system operation.
- **Power Supply B** — Connection to a second power source is optional to provide redundancy.

For more information about the Cuda 12000 cooling and power features, see the "Cuda 12000 IP Access Switch Installation Guide."

Two DB-15 connectors — *alarms in* and *alarms out* — on the rear of the Cuda 12000 enable communication of alarms from fan trap and power supply units. The following table describes the DB-15 connectors:

Table 8-1 DB-15 Connectors

Connector	Description
Alarms In	Receives fault signals from the connected units.
Alarms Out	Transmits fault signals to an external device.

For information about connecting the DB-15 connectors to the auxiliary devices, refer to the Cuda 12000 IP Access Switch Installation Guide.

You can configure reporting for a number of fault conditions, such as fan rotation and temperature, power alarms and backplane temperature. Reception of a fault signal from the device results in an SNMP trap or syslog message, which is sent to the specified destinations. To view the SNMP trap and syslog message destinations on the system use the **show snmp notify** command. (Refer to the next chapter, "Simple Network Management Protocol (SNMP)" on page 161 for information about configuring destinations for fault events.)

Configuring power and fan tray fault reporting involves performing the following tasks:

- You must specify whether the auxiliary device utilizes an active-high or active-low assertion level to report fault conditions, as described in "Assertion Levels" on page 150.
- Configure the faults that you want the system to report, as described in "Configuring Fault Reporting" on page 153.
- Specify the alarms that you want to send over the *alarms out* DB-15 connector to an external indication device, as described in "Configuring Alarms Out" on page 157.

Assertion Levels

When a fault condition occurs on the fan unit or power supply, a signal is sent to TAC indicating a fault condition. The signal that the fan unit or power supply sends may use one of the following assertion levels:

- **Active-High** — Signal indicates the assertion state as a logic ONE state.
- **Active-Low** — Signal indicates the assertion state as a logic ZERO state.

The power supply and fan unit assertion levels are configured for the following parameters (see Table 8-2). Assertion levels are normally set to active-low, unless otherwise specified by the auxiliary device vendor. The default is set to active-low.

Table 8-2 Assertion Level Parameters

Parameter	Description
PS-temp	Set to report the temperature of the power supply.
DC-monitor	Set to report DC current faults of the power supply.
AC-monitor	Set to report AC current faults of the power supply.
Fan-temp	Set to report temperature faults of the fan unit.
Fan-rotation	Set to report rotation faults of the fan unit.

Configuring the Power Assertion Level

You must verify the assertion level specified by the power supply vendor to indicate fault conditions, and set the assertion levels as specified by the vendor.

To configure the assertion level that the power supply utilizes when indicating a fault condition, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Set the assertion level to report AC current faults.	**aux-device ac-monitor fault-level** {**active-high** | **active-low**}
3. Set the assertion level to report DC current faults.	**aux-device dc-monitor fault-level** {**active-high** | **active-low**}
4. Set the assertion level to report power supply temperature faults.	**aux-device ps-temp fault-level** {**active-high** | **active-low**}

To display the assertion levels currently configured for both AC and DC and power supply temperature fault indications, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Display the assertion level currently configured for the reporting of AC current faults.	**show aux-device ac-monitor**
3. Display the assertion level currently configured for the reporting of DC current faults.	**show aux-device dc-monitor**
4. Display the assertion level currently configured for the reporting of power supply temperature faults.	**show aux-device ps-temp**

Configuring Fan Unit Assertion Levels

The fan unit, by default, sends an active-low signal to TAC to report fan temperature and rotation faults. You configure the fan unit to send an active-high signal if so specified by the fan unit vendor.

To configure the fan unit assertion level, perform the following tasks:

Task	Command
1. Enter configuration mode.	**root**
2. Set the assertion level of the signal used to report fan temperature faults to the management module.	**aux-device fan-temp fault-level** {**active-high** | **active-low**}
3. Set the assertion level of the signal used to report fan rotation faults to the management module.	**aux-device fan-rotation fault-level** {**active-high** | **active-low**}

To display the fan unit assertion levels currently configured for both fan temperature and rotation fault indications, perform the following tasks:

Task	Command
1. Enter configuration mode.	**root**
2. Display the assertion level currently configured for the reporting of fan temperature faults.	**show aux-device fan-temp**
3. Display the assertion level currently configured for the reporting of fan rotation faults.	**show aux-device fan-rotation**

Example

The following example displays the assertion level currently configured for the fan temperature:

```
cli:192.168.208.3:root# show aux-device fan-temp
Assert Fan Temp Fault            active-high
```

Configuring Fault Reporting

The system reports faults in the form of SNMP traps and syslog messages. You must configure the faults for which you want to be notified. For each fault that you choose to report, the system sends an SNMP trap or syslog message to all specified destinations if a fault is detected. SNMP traps and syslog messages are also sent when there is a state transition from *okay* to *faulted* or a transition from *faulted* to *okay*.

The Cuda 12000 allows you to report on the following fault conditions.

Table 8-3 Fault Conditions

Fault	Description
backplane	A payload blade asserted a backplane system fault condition.
backplane-power	One or more payload blades detected an internal Power Fault.
backplane-power-a	One or more payload blades detected a Power_A (48V) Fault.
backplane-power-b	One or more payload blades detected a Power_B (48V) Fault.
backplane-temp	One or more payload blades detected a Temperature Fault.
bits-a	The chassis manager associated with the chassis detected a loss of the BITS-A clock.
bits-b	The chassis manager associated with the chassis detected a loss of the BITS-B clock.
blue	One or more payload blades has asserted a Blue Alarm.
fan-rotation	The fan tray associated with the chassis detected one or more non-rotating fans.
fan-temp	The fan tray associated with the chassis detected an inlet temperature > 50 deg. C.
local-pwr-a	The chassis manager associated with the chassis detected a loss of Power_A (48V).
local-pwr-b	The chassis manager associated with the chassis detected a loss of Power_B (48V).
processor-temp	The chassis manager associated with the chassis detected a processor over-temperature condition.

Fault	Description
ps-ac	The power supply associated with the chassis detected the loss of one or more AC inputs.
ps-dc	The power supply associated with the chassis detected a DC out-of-range fault.
ps-temp	The power supply associated with the chassis detected an over-temperature condition.
red-alarm	One or more payload blades has asserted a Red Alarm.
yellow alarm	One or more payload blades has asserted a Yellow Alarm.

To configure the faults for which you want to be notified, perform the following tasks. By default, the fault reporting status is "disabled" for each fault:

Task	Command
1. Enter root mode.	**root**
2. Specify the faults that you want reported.	**chassis-fault** {**backplane** | **backplane-power** | **backplane-power-a** | **backplane-power-b** | **backplane-temp** | **bits-a** | **bits-b** | **blue** | **fan-rotation** | **fan-temp** | **local-pwr-a** | **local-pwr-b** | **processor-temp** | **ps-ac** | **ps-dc** | **ps-temp** | **red-alarm** | **yellow**}

Removing a Fault Notification

In the event that you no longer wish to be notified of a fault condition, you may remove a specified fault notification by performing the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Remove the fault condition from the notification report.	**no chassis-fault** {**backplane** | **backplane-power** | **backplane-power-a** | **backplane-power-b** | **backplane-temp** | **bits-a** | **bits-b** | **blue** | **fan-rotation** | **fan-temp** | **local-pwr-a** | **local-pwr-b** | **processor-temp** | **ps-ac** | **ps-dc** | **ps-temp** | **red-alarm** | **yellow**}

Viewing Fault Reporting Status

Each fault condition displays one of the following states.

- **disabled** — Reporting is not specified for the fault condition. An SNMP Trap or syslog message is not generated when the fault condition occurs.
- **faulted** — Fault reporting is specified. An SNMP trap or syslog message is generated when the fault condition occurs.
- **okay** — Fault reporting is specified.

Use the **show chassis-fault status** command to display the state of each fault condition.

Example

The following example displays that faults are to be reported for the backplane power and local power-source-a conditions. In addition, the display indicates that faults have not occurred for the specified conditions.

```
cli:192.168.244.212:root# chassis-fault backplane-power
local-pwr-a
cli:192.168.244.212:root# show chassis-fault status

Chassis Fault Status

Bits A Fault                          disabled
Bits B Fault                          disabled
Backplane System Fault                disabled
Backplane Temp Fault                  disabled
Backplane Power Fault                     okay
Backplane Power A Fault               disabled
Backplane Power B Fault               disabled
Red Alarm Fault                       disabled
Blue Alarm Fault                      disabled
Yellow Alarm Fault                    disabled
Processor Temp Fault                  disabled
Ps Temp Fault                         disabled
Ps AC Fault                           disabled
Ps DC Fault                           disabled
Fan Temp Fault                        disabled
Fan Rotation Fault                    disabled
Local Pwr A Fault                         okay
Local Pwr B Fault                     disabled
```

Configuring Alarms Out

A DB-15 connector on the Cuda 12000 chassis rear panel serves as the *alarms out* port. You can configure the Cuda 12000 to send specific types of alarm signals out this DB-15 connector to an external indication device to notify the external device that a particular type of fault has occurred. (Refer to the *Cuda 12000 IP Access Switch Installation Guide* for information about cabling the DB-15 connector.)

Each fault can generate one or more types of alarm signals. Table 8-4 shows the alarm signals that you can send over the *alarms out* port and the associated faults that you can use to trigger these signals:

Table 8-4 Alarm Signals and Associated Faults

You Can Configure This Signal:	To Provide Notification of These Faults:
Temp Alarm	■ backplane-temp-fault ■ processor-temp-fault ■ ps-temp-fault ■ fan-temp-fault
Sys Alarm	■ backplane-system-fault ■ backplane-temp-fault ■ backplane-pwr-fault ■ local-pwr-a-fault ■ local-pwr-b-fault ■ red-alarm-fault ■ ps-temp-fault ■ ps-ac-fault ■ ps-dc-fault ■ fan-temp-fault ■ fan-rotation-fault
Red Alarm	■ bits-a-fault ■ bits-b-fault ■ red-alarm-fault
Blue Alarm	■ blue-alarm-fault
Yellow Alarm	■ yellow-alarm-fault

You Can Configure This Signal:	To Provide Notification of These Faults:
Power Alarm	■ local-pwr-a-fault ■ local-pwr-b-fault ■ backplane-pwr-fault ■ backplane-pwr-a-fault ■ backplane-pwr-b-fault ■ ps-ac-fault ■ ps-dc-fault
Temp Fault	■ processor-temp-fault ■ ps-temp-fault ■ fan-temp-fault
Power Fault	■ local-pwr-a-fault ■ local-pwr-b-fault ■ backplane-pwr-a-fault ■ backplane-pwr-b-fault ■ ps-ac-fault ■ ps-dc-fault
PowerA Fail	■ local-pwr-a-fault
PowerB Fail	■ local-pwr-b-fault
Clock	■ bits-a-fault ■ bits-b-fault ■ red-alarm-fault

To configure the alarm signals for transmit over the DB-15 connector and the associated signals that can trigger them, perform the following tasks. By default, all faults are configured to *not* send out their associated alarm signals over the DB-15 connector:

Task	Command
1. Enter root mode.	**root**
2. Specify the types of alarm signals that you want to enable, to set the fault to send alarm signals out the DB-15 connector.	**no aux-device db15 alarm** {**blue |** **clock** [**bits-a**] [**bits-b**] [**red-alarm**] **|** **power-alarm** [**backplane-power**] [**backplane-pwr-a**] [**backplane-pwr-b**] [**local-pwr-a**] [**local-pwr-b**] [**ps-ac**] [**ps-dc**] **|** **power-fail-A |** **power-fail-B |** **red** [**bits-a**] [**bits-b**] [**red-alarm**] **|** **system** [**backplane**] [**backplane-power**] [**backplane-temp**] [**fan-rotation**] [**fan-temp**] [**local-pwr-a**] [**local-pwr-b**] [**ps-ac**] [**ps-dc**] [**ps-temp**] [**red-alarm**] **|** **temp** [**backplane-temp**] [**fan-temp**] [**processor-temp**] [**ps-temp**] **|** **yellow**}
3. Specify the types of alarm signals that you want to disable, to prohibit the faults from being sent out the DB-15 connector.	**aux-device db15 alarm** {**blue |** **clock** [**bits-a**] [**bits-b**] [**red-alarm**] **|** **power-alarm** [**backplane-power**] [**backplane-pwr-a**] [**backplane-pwr-b]** [**local-pwr-a**] [**local-pwr-b**] [**ps-ac**] [**ps-dc**] **|** **power-fail-A |** **power-fail-B |** **red** [**bits-a**] [**bits-b**] [**red-alarm**] **|** **system** [**backplane**] [**backplane-power**] [**backplane-temp**] [**fan-rotation**] [**fan-temp**] [**local-pwr-a**] [**local-pwr-b**] [**ps-ac**] [**ps-dc**] [**ps-temp**] [**red-alarm**] **|** **temp** [**backplane-temp**] [**fan-temp**] [**processor-temp**] [**ps-temp**] **|** **yellow**}

Viewing Alarm Signals Out the DB-15 Connector

You may display the current configuration of the alarm signals over the DB-15 connector. To do this, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Display current configuration of the alarm signals out the DB-15 connector.	**show aux-device db15**

Example

In the following example:

- A Temp Alarm, Sys Alarm, Red Alarm, Yellow Alarm and Power Alarm signal have been configured in the past to be sent out the DB-15 connector for all associated faults.
- The administrator configures a Blue Alarm signal to not be sent out the DB-15 connector for all associated faults.

```
cli:192.168.208.3:root# aux-device db15 alarm blue
cli:192.168.208.3:root# show aux-device db15
Temp Alarm                backplane-temp-fault              enabled
                          processor-temp-fault              enabled
                          ps-temp-fault                     enabled
                          fan-temp-fault                    enabled
Sys Alarm                 backplane-system-fault            enabled
                          backplane-temp-fault              enabled
                          backplane-power-fault             enabled
                          local-pwr-a-fault                 enabled
                          local-pwr-b-fault                 enabled
                          red-alarm-fault                   enabled
                          ps-temp-fault                     enabled
                          ps-ac-fault                       enabled
                          ps-dc-fault                       enabled
                          fan-temp-fault                    enabled
                          fan-rotation-fault                enabled
Red Alarm                 bits-a-fault                      enabled
                          bits-b-fault                      enabled
                          red-alarm-fault                   enabled
Blue Alarm                blue-alarm-fault                  disabled
Yellow Alarm              yellow-alarm-fault                enabled
Power Alarm               local-pwr-a-fault                 enabled
```

9 SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

Simple Network Management Protocol (SNMP) is a standard for managing networks. This chapter provides an overview of SNMP (refer to "About SNMP" on page 162). This chapter also provides information on performing the following tasks to configure SNMP on the Cuda 12000:

- Configuring SNMP Access Control (page 164)
- Configuring System Name, Contact, and Location (page 180)
- Configuring SNMP Event Notification Types (page 182)
- Monitoring SNMP (page 196)

Sample SNMP configurations are provided at the end of this chapter. These samples illustrate how to set up complete SNMPv1, SNMPv2, and SNMPv3 security schemes. Refer to "Sample SNMP Configurations" on page 198 for more information.

About SNMP

This section provides an overview of SNMP. For further information about the SNMP protocol, refer to the RFCs listed below or other general publications specific to SNMP.

SNMP is a network management protocol that provides a standard for network management systems. In the SNMP scheme, a network management system contains two primary components: a manager and agents. The manager is the workstation or console where the network administrator performs network management functions. Agents are entities that interface with the device being managed. The Cuda 12000 runs an SNMP agent.

Devices, such as the Cuda 12000, that are managed using SNMP contain managed objects, such as configuration parameters and performance statistics. These objects are defined in a management information base (MIB). SNMP allows managers and agents to communicate for the purpose of accessing MIB objects.

SNMP provides access control to MIB objects, which defines who can access MIB objects and their associated access privileges. In SNMPv1 and v2c, access control is managed through associations of agents and managers called communities and security groups. In SNMPv3, access control is managed by the user and context, and an associated security group. Refer to "Configuring SNMP Access Control" on page 164 for more information on access control.

SNMP security is defined by Security Models and Security Levels. A combination of a security model and a security level determines which security mechanism is used when handling an SNMP packet. A security model may authenticate messages by providing data integrity, data origin authentication, data confidentiality, message timeliness and limited replay protection. A security model is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. The level of security is determined primarily by the specific SNMP application implementation and by the specific security model implementation.

The Cuda 12000 supports *DOCSIS 1.1 OSS Interface Specification (SP-OSSIv1.1-I02-000714)* and SNMP configuration, as defined by RFCs 1157, 2571, 2572, 2573, 2574, 2575 and 2576.

Configuring and monitoring SNMP on the Cuda 12000 involves the following processes. These processes are explained in the sections that follow:

1. Configuring SNMP access control.
2. Configuring system name, contact, and location information. Refer to "Configuring System Name, Contact, and Location" on page 180.
3. Configuring event notification. Refer to "Configuring SNMP Event Notification Types" on page 182.
4. Monitoring SNMP. Refer to "Monitoring SNMP" on page 196.

Configuring SNMP Access Control

SNMP Access Control defines how SNMP will controls access to MIB objects.

In SNMP versions 1 and 2c, access control is configured by a community-based model. A community associates an SNMP agent and an SNMP management application. You assign a name to the community, and the agent and management application use this name to authenticate SNMP messages exchanged between them.

In SNMP version 3, access control is configured by a user-context model. These models are described in the sections that follow.

Configuring access control for MIB views and groups is common to all versions.

SNMP access control follows a hierarchy and it is recommended that you perform configuration functions in the following order:

1. Configure SNMP Access Views.
2. Configure SNMP Groups.
3. Configure Models, as follows:
   - **a** Configure SNMPv1, v2c Communities, and/or
   - **b** Configure SNMPv3 Users and Contexts

Configuring SNMP Access Views

SNMP Access Views control access to a MIB subtree. Configuring SNMP Access Views involves the following:

1. Creating a MIB view. You create a MIB view by specifying a name for the view, by defining the MIB subtree to be viewed, and by specifying whether instances of the MIB subtree are included in the MIB view or excluded from the MIB view.
2. Specifying the storage type for the view.
3. Specifying the status of the view.

The following table describes the parameters that you set to configure SNMP Views:

Table 9-1 Parameters Associated with SNMP View Configuration

Parameter	Description
View Name	The name of the MIB view.
Subtree	The MIB subtree that defines the family of view subtrees.
Type	Indicates whether the instances of the MIB subtree are included or excluded from the MIB view.
Storage	Indicates how the MIB view is stored.The options are: ■ volatile: The entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default): The entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent: The entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly: The entry is stored in non-volatile memory. You cannot delete or modify the information.
Status	Indicates whether the MIB view is Active or Not in Service. If status is set to Enable, the MIB view is Active. If status is set to Disable, the MIB view is not in service.

Perform the following tasks to configure SNMP Access Views:

Tasks	Commands
1. Enter configuration mode.	**root**
2. Create a MIB view by performing the following tasks: ■ Specify the name of the view. ■ Specify the MIB subtree that defines the family of views. You can enter the MIB value as an Object Identifier (OID), and OID with wildcards, or an OID name description. ■ Set the corresponding instances of the MIB subtree to be included or excluded in the MIB view.	**snmp-server view** <view name> <oid-tree> {**included** | **excluded**}
3. Specify the storage type for the MIB view. By default, storage type is set to NonVolatile.	**snmp-server view** <view name> <oid-tree> {**included** | **excluded**} [**storage {volatile** | **nonvolatile** | **permanent** | **readonly**}]
4. Set the status of the MIB view. You enable or disable the status. Enable sets the status to Active. Disable sets the status to Not in Service. By default, the status is set to Active.	**snmp-server view** <view name> <oid-tree> {**included** | **excluded**} [**status** {**enable** | **disable**}]
5. Display an SNMP MIB view.	**show snmp view** [<view name>]
6. Remove an SNMP MIB view.	**no snmp-server view** <view name>

Example

The following example configures and displays an SNMP MIB view using the default storage type and status.

```
cli:192.168.208.3:root# snmp-server view auditorview1 1.3.6.1 included
cli:192.168.208.3:root# show snmp view

row count: 5

View Name        Subtree                     Type     Storage     Status
---------------- --------------------------- -------- ----------- -----------
public           1.3.6.1                     Included NonVolatile Active
private          1.3.6.1                     Included NonVolatile Active
guitraps         1.3.6.1                     Included NonVolatile Active
v1default        1.3.6.1                     Included NonVolatile Active
auditorview1     1.3.6.1                     Included NonVolatile Active
```

Configuring SNMP Groups

SNMP groups restrict read, write and notify access to certain parts of the MIB. Configuring SNMP groups involves:

1. Creating a group.
2. Assigning the group a security mode and security level to process SNMP messages.
3. Specifying how the group is stored and assigning the group access privileges to an SNMP MIB view.

You set the following parameters to configure SNMP Groups:

Table 9-2 SNMP Group Configuration Parameters

Parameter	Description
Group	The name of the SNMP group for the SNMP entity.
Context	The name of the context associated with the specific group.
Model	The SNMP security model used to process SNMP messages and gain access to the group. You can choose V1, V2c or V3.
Level	The level of security to process SNMP messages. You can choose one of the following three levels: ■ No Authentication (noauth): Provides no authentication and no encryption. This is the lowest level of security. V1 and V2c security models provide only this level of security. ■ Authentication (auth): Provides authentication but no encryption. Only V3 security model provides this level of security. ■ Privacy (priv): Provides authentication and encryption. This is the highest level of security. Only V3 security model provides this level of security.
Read View	Authorizes the group to have read access to the specific MIB view.
Write View	Authorizes the group to have write access to the specific MIB view.
Notify View	Authorizes the group to have notify access to the specific MIB view.

Parameter	Description
Storage	Specifies how the group entry is stored: ■ volatile: The entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default): The entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent: The entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly: The entry is stored in non-volatile memory. You cannot delete or modify the information.

Perform the following tasks to configure an SNMP group. Refer to the configuration examples below:

Read, write or notify privileges are associated to an SNMP MIB view. If an SNMP view already exists, you assign the privileges to that existing view name. If an SNMP view does not already exist, you can create a view name for a new view when you assign the access privileges.

Task	Command
1. Enter configuration mode.	**root**
2. Create an SNMP group. When an SNMP group is created, by default, the Read View name is assigned as v1default	**snmp-server group** <group name> {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}}
3. Assign the group's access to an SNMP view. If an SNMP view does not already exist, you assign a name for a new view.	**snmp-server group** <group name> {**v1 | v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** <view name>] [**write** <view name>] [**notify** <view name>]
4. Associate an existing context to the group.	**snmp-server group** <group name> {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context** <context name>]
5. Specify the storage type for this group.	**snmp-server group** <group name> {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**storage** {**volatile** | **nonvolatile** | **permanent** | **readonly**}]

Task	Command
6. Display SNMP group information.	**show snmp group** [<group-name>]
7. Remove an SNMP group.	**no snmp-server group** <group-name>

Example 1

The following example creates a new group:

```
root# snmp-server group alemap v3 auth
root# show snmp group alemap

row count: 18

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- -----------
alemap             V3    Auth   v1default                          NonVolatile
```

The following example assigns the group read privileges to an SNMP MIB view:

```
root# snmp-server group alemaps v1 read adc
root# show snmp group

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- -----------
alemaps            V1    NoAuth adc                                NonVolatile

cli:192.168.208.3:root#
```

The following example associates an existing context to the group:

```
root# snmp-server group alemaps v1 read public context adc
root# show snmp group

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- -----------
alemaps    adc     V1    NoAuth public                              NonVolatile
```

Example 2

The following example specifies the storage type for a group:

```
root# snmp-server group hms v3 auth storage volatile
root# show snmp group

Group      Context Model Level  Read View   Write View  Notify View Storage
---------- ------- ----- ------ ----------- ----------- ----------- -----------
hms                V3    Auth   v1default                          Volatile
```

Configuring SNMP Communities

SNMP versions 1 and 2c use a community to control access to a MIB object. A community is a pairing relationship between an SNMP agent and an SNMP application. The network administrator assigns the community a name. The community assigns specific rights and privileges to authenticate SNMP messages. The community passes on the messages to an associated group.

You set the following parameters to configure SNMP Communities:

Table 9-3 Parameters contained in SNMP Community Configuration

Parameter	Description
Name	The name assigned to the SNMP community. An SNMP community name may contain up to 32 alphanumeric characters.
Security Name	The name assigned to security group for the associated SNMP community. A security group name may contain up to 32 alphanumeric characters
IP Address	The IP address of a host that is a member of the SNMP community. If you do not specify an IP address, all hosts are allowed access using the community string.
Mask	The mask for the specified IP address. The mask allows you to specify a range of hosts. For example, you can specify an IP address of 220.220.0.0 with a mask of 255.255.0.0. This IP-mask address combination allows any host from 220.220.0.0 through 220.220.255.255 to access MIB objects in the specified SNMP community.
Context	The name of the SNMP context that is used with the specified community when accessing the security group. The context is one of the parameters that allows access to a group entry, along with group name, security model and security level. *NOTE: Refer to the section "Configuring SNMPv3 Contexts" on page 178, for detailed information about SNMP contexts.*

Parameter	Description
Storage	Indicates how the SNMP community's attributes are stored. The options are: ■ volatile: The entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default): The entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent: The entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly: The entry is stored in non-volatile memory. You cannot delete or modify the information.

Perform the following tasks to configure an SNMPv1 or v2c community. Refer to the configuration example below:

Task	Command
1. Enter configuration mode.	**root**
2. Create an SNMP community and a corresponding security group.	**snmp-server community** <community name> <security name>
3. Specify the host that has access to the SNMP community.	**snmp-server community** <community name> <security name> **address** <ip-address>
4. Specify the Mask address for a range of hosts.	**snmp-server community** <community name> <security name> **address** <ip-address> **mask** <mask address>
5. Name the SNMP context to be used with the SNMP community.	**snmp-server community** <community name> <security name> [**address** <ip-address> [**mask** <mask address>]] [**context** <context>]
6. Display a specific SNMP community or all SNMP communities.	**show snmp community** [<community name>]
7. Remove an SNMP community and its security group.	**no snmp-server community** <community name>

Example

The following example creates and displays a specific SNMP community:

```
cli:root# snmp-server community beta build address 192.168.20.12 mask
255.255.255.0 context cuda
cli:192.168.208.3:root# show snmp community

row count: 5

Name        Security Name           Context             Storage
-------------------------------------------------------------------
bat         all                                         NonVolatile
beta        build                   cuda                NonVolatile
guitraps    guitraps                                    NonVolatile
private     adc                     adc                 NonVolatile
public      adc                     adc                 NonVolatile
```

Configuring SNMPv3 Users

The SNMPv3 user is anyone who requires management operations to be authorized by a particular SNMP entity. SNMP entities must have knowledge of a user and the user's attributes.

Configuring an SNMPv3 user involves the following:

1. Specifying a user's name.
2. Specifying the user's security attributes for an SNMP entity.
3. Specifying the storage type for the user.
4. Specifying the status of the user.

The following table describes the parameters that you set to configure SNMP users:

Table 9-4 Parameters Contained in SNMPv3 User Configuration

Parameter	Description
Name	The name assigned to the user for the SNMP entity. A user name may contain up to 32 alphanumeric characters.
Authentication	The security attribute for this user that indicates whether messages sent on behalf of this user can be authenticated. Authentication is defined by two protocols: HMC-MD5-96 and HMAC-SHA-96.
Privacy	Indicates whether messages sent on behalf of this user will be protected from disclosure. Privacy is defined by the CBC-DES Symmetric Encryption Protocol.

Parameter	Description
Storage	Indicates how the user's attributes are stored. The options are: ■ volatile: The entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default): The entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent: The entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly: The entry is stored in non-volatile memory. You cannot delete or modify the information.
Status	Indicates whether the user is active or not in server active. Set to Enable to indicate Active, or set to Disable to indicate Not in Service.

Perform the following tasks to configure an SNMP user. An example of an SNMPv3 user configuration is displayed below:

Tasks	Commands
1. Enter configuration mode.	**root**
2. Create a user for the SNMP entity.	**snmp-server user** <user>
3. Specify the authentication type for this user. You must specify authentication type in order for messages to be authenticated. By default, authentication type is None.	**snmp-server user** <user> [**auth** {**md5 | sha**} <auth-password>]
Optionally, specify the **priv des56** argument to protect messages from disclosure. By default, messages are not protected from disclosure.	**snmp-server user** <user> [**auth** {**md5 | sha**} <auth-password>] [**priv des56** <priv-password>]]

Tasks	Commands
4. Specify how user attributes are stored. By default, storage type is set to NonVolatile.	**snmp-server user** <user> [**storage** {**volatile | nonvolatile | permanent | readonly**}]
5. Specify the user's status. By default, the Cuda sets status type to Active.	**snmp-server user** <user> [**status** {**enable | disable**}]
6. Display SNMP user attributes.	**show snmp user** [<user>]
7. Remove an SNMP user.	**no snmp-server user** <user>

Example

The following example configures an SNMP user with authentication type and privacy attributes:

```
root# snmp-server user mapale auth sha 000111 priv des56 000111
root# show snmp user mapale

Name                            Authentication Privacy Storage     Status
------------------------------- -------------- ------- ----------- ------------
mapale                          HMAC-SHA-96    CBC-DES NonVolatile Active
cli:192.168.208.3:root#
```

Configuring SNMPv3 Contexts

SNMPv3 uses contexts to control access to a MIB object. A context is a collection of management information that is accessed by an SNMP entity. A single SNMP entity may be in more than one context. A single SNMP entity may have access to many contexts.

Configuring SNMPv3 contexts involves:

1. Creating a context.
2. Setting a storage type for the context.
3. Setting the status for the context.

The following table describes the parameters that you set to configure SNMPv3 Contexts:

Table 9-5 Parameters Contained in SNMPv3 Context Configuration

Parameter	Description
Context Name	The name that identifies a context. A null value indicates a default context. A context name may include up to 32 alphanumeric characters.
Storage	Indicates how the context entry is stored. Following is a list of the storage types. By default, storage is set to NonVolatile:
Non-volatile	The entry is stored in non-volatile memory. The information is not lost during a system reboot.
Permanent	The entry is stored in non-volatile memory. You cannot delete the information but you can make modifications.
Read-only	The entry is stored in non-volatile memory. You cannot delete or modify the information.
Volatile	The entry is stored in volatile memory. The information is lost during a system reboot.
Status	Sets the state of the context, as follows:
Enable	Activates a context.
Disable	Temporarily sets the context to "Not In Service." By default, the context is enabled.

Perform the following tasks to configure SNMPv3 contexts. Refer to the configuration example below:

Task	Command
1. Enter configuration mode.	**root**
2. Provide the name of the context. Enter a single text or numeric string, up to 32 characters.	**snmp-server context** <context name>
3. Set the storage type for the context. By default, storage is set to NonVolatile.	**snmp-server context [storage** {**nonvolatile | permanent | readonly | volatile**}]
4. Set the status for the context. By default, status is set to enable.	**snmp-server context** [**status** {**enable | disable**}]
5. Display all context information for an SNMP entity.	**show snmp context** [<context name>]
6. Remove the context information.	**no snmp-server context** <context name>

Example

The following example displays the configuration for the context "tech" and displays all contexts for an SNMP entity.

```
root# snmp-server context tech storage permanent status enable
root# show snmp context

row count: 3

Name                            Storage      Status
------------------------------- ------------ ------------
adc                             NonVolatile  Active
june                            NonVolatile  Active
tech                            Permanent    Active

cli:192.168.208.3:root#
```

Configuring System Name, Contact, and Location

For the Cuda 12000, you can configure system name, contact, and location information. This information is stored in the sysName, sysContact, and sysLocation MIB variables. This information is described as follows:

Table 9-6 System Name, Contact, and Location Parameters

Parameter	Description
Name	The name of the system (sysName MIB object).
Contact	The type of contact for this network (sysContact MIB object). The contact is typically a network administrator's name, extension, and/or e-mail address.
Location	The physical location of the Cuda 12000 (sysLocation MIB object).

Perform the following tasks to configure name, contact, and location information for the Cuda 12000:

A name, contact, or location text string may contain up to 255 alphanumeric characters. If a text string contains spaces, you may enclose the string in quotes.

Task	Command
1. Enter configuration mode.	**root**
2. Specify the SNMP contact.	**snmp-server contact** <contact>
3. Specify the system's name.	**snmp-server name** <name>
4. Specify the system's physical location.	**snmp-server location** <location>
5. Display the contact, name and location information.	**show snmp**
6. Remove the contact, name or location information.	**no snmp-server** {**contact | name | location**}

Example

The following example creates and displays name, contact, and location information:

```
root# snmp-server name "cuda 111"
root# snmp-server contact “John Smith, x334”
root# snmp-server location "bldg. 1400"
root# show snmp
Contact                    John Smith, x334
Name                       cuda 111
Location                   bldg. 1400
SNMP packets received                182168
Bad SNMP version errors                   0
Unknown community names                   0
Illegal community names                   0
Encoding errors                           0
Silent drops                              0
Unknown security models                   0
Invalid messages                          0
Unknown PDU handlers                      0
Authentication traps                 disable

cli:192.168.208.3:root#
```

Configuring SNMP Event Notification Types

Notifications indicate that a system event occurred, such as a physical fault that affects the chassis, and system faults that may impact the operation of the management module or any of the application modules.

Notifications are sent to an SNMP host. The SNMP host may be the default local host on the management module, or an external host that you configure to receive the notifications.

The local host is the default host that is pre-configured and shipped with your chassis.Notifications, for the local host, are sent to IP address 127.0.0.1. If CudaView is installed on the chassis, CudaView uses the local host to display notifications. The local host IP address should not be changed.

Defining event notification involves configuring which notifications you want to send to the SNMP host and how to send the notifications to the SNMP host. Notifications may be sent as traps or informs.

Traps are notifications that are not acknowledged by the SNMP manager, so they are considered unreliable. In addition, traps are not held in memory. Informs are notifications that are acknowledged by the SNMP manager, so they are considered reliable. If an inform is sent and not acknowledged, it may be sent again. Informs are held in memory, which means they consume more router and network resources.

The following table lists the system events and their associated Event Classes. For more information about Event Classes, refer to Chapter 10, *Managing System Events,* on page 203.

Table 9-7 List of System Events

System Event	Description	Event Class
Cluster events:	Cluster events refer to faults that affect the management module.	
■ authentication-failure	SNMP receives a bad Community Name.	Notice
■ bcm-failover-down	Services are going down. This notification type applies to redundant configurations only.	Notice
■ bcm-failover-up	Services are coming up. This notification type applies to redundant configurations only.	Notice
■ bcm-state-change	A change in the craft port IP address.	Notice
■ bcm-sw-mismatch	The secondary will not come up because its software revision does not match the software revision of the primary.	Notice
■ trace-log	For ADC internal use only.	Notice
■ cold-start	Generated when module boots from power up.	Notice
■ warm-start	Generated when module boots from reset.	Notice
■ icl-state-change	A change in the ICL link.	Notice
Module events:	Module events refer to hardware faults that affect any application module.	
■ cable-modem-auth-failure	Cable modem failed authorization and did not register.	Notice
■ cable-modem-down	Cable modem is not operational.	Critical
■ cable-modem-up	Cable modem is operational.	Notice
■ card-down	Application module failure.	Critical
■ card-up	Application module is operational.	Notice
■ dhcp-relay-not-configured	DHCP configuration error.	Warning
■ local-sonet-alarm	A transmission problem is detected from the transmitter.	Error
■ remote-sonet-alarm	A transmission problem is detected from the receiver.	Error

System Event	Description	Event Class
■ Interface-related events:	Interface-related events refer to faults that affect the link state of the interface.	Notice
■ link up	Link to IP network is operational.	Notice
■ link down	Link to IP network is not operational.	Error
■ chassis-fault	Auxiliary device-related event that refers to faults associated to the fan tray, power source and clock sources. *NOTE: Reference "Timing and Alarm Module Fault Reporting" on page 155, for a list of the chassis-related notification types*.	Critical
■ chassis-fault-cleared	Indicates the chassis event that caused a fault is fixed.	Notice
Provisioning events:	Provisioning events refer to faults that pertain to the FastFlow BPM running on the Cuda 12000.	
■ duplicate-addr	A duplicate IP address has been detected.	Notice
■ isp-addr-high	The free IP address count exceeded the upper threshold for the specified ISP.	Notice
■ isp-addr-low	The free address count fell below the lower threshold for the specified ISP.	Notice
■ ldap-failed	A directory server access failure occurred.	Notice
■ ldap-restored	Directory server access is operational after a failure.	Notice
■ prov-service	A FastFlow BPM service started, stopped, or failed.	Notice
■ subnet-addr-high	The free IP address count exceeded the high available address threshold for a subnet.	Notice
■ subnet-addr-low	The free IP address count fell below the low available address threshold for a subnet.	Notice
DOCSIS events:	DOCSIS events refer to initialization faults on DOCSIS and EuroDOCSIS modules.	
■ docs-dyn-rsp-fail	A dynamic service response failure occurred during the dynamic services process.	Warning

System Event	Description	Event Class
■ docss-dyn-ack-fail	A dynamic service acknowledgement failure occurred during the dynamic services process.	Warning
■ docs-dyn-req-fail	A dynamic service request failure occurred during the dynamic services process.	Warning
■ docs-bpi-init	A BPI initialization attempt failure occurred during the registration process.	Informational
■ docs-bpkm	A baseline privacy key management operation failed.	Error
■ docs-dcc-ack-fail	A dynamic channel change acknowledgement failed during the dynamic channel change process in the CMTS.	Warning
■ docs-dcc-req-fail	A dynamic channel change request failed during the dynamic channel change process in the cable modem and was detected by the CMTS.	Warning
■ docs-dcc-rsp-fail	A dynamic channel change response failed during the dynamic channel change process in the CMTS.	Warning
■ docs-dynamic-sa	A dynamic security association failed.	Warning
■ docs-init-ack-fail	A registration acknowledgement failure from the cable modem occurred during the cable modem initialization process and was detected on the CMTS side.	Warning
■ docs-init-req-fail	A registration request failure from the cable modem occurred during the cable modem initialization process and was detected on the CMTS side.	Warning
■ docs-init-rsp-fail	A registration response failure from the cable modem occurred during the cable modem initialization process and was detected on the CMTS side.	Warning
Routing events:	Routing events refer to events that indicate a change in the state of OSPF neighbors and OSPF virtual neighbors.	

System Event	Description	Event Class
■ ospf-nbr-state	Signifies a change in the state of an OSPF neighbor on a physical interface. To send this notification type, note that you also have to enable sending of OSPF neighbor state traps using the **report** command.	Notice
■ ospf-virt-nbr-state	Signifies a change in the state of an OSPF neighbor on a virtual interface. To send this notification type, note that you also have to enable sending of OSPF virtual neighbor state traps using the **report** command.	Notice
Modem deregistration event:	A modem deregistration event refers to the deregistration of cable modems.	
■ dereg-modems	Signifies that a number or percentage of modems have deregistered over the deregistration time interval.	Warning

Configuring event notification types involves:

1. Defining an SNMP host to receive trap messages.
2. Specifying the UDP port number on which the SNMP host will receive traps.
3. Specifying the maximum message size (MMS) that the SNMP entity will transmit or receive, and process.
4. Specifying how the definition of the SNMP host that will receive events is stored.
5. Specifying the events for which you want to receive notification, and specifying how to send the notifications.

The following table describes the parameters that you set to configure notifications:

Table 9-8 Parameters Contained in Event Notification Configuration

Parameter	Description
Host:Port	The IP address and UDP port number on which the SNMP host is configured to receive traps. The UDP port range is 1 to 65535 and the default is 162.
Timeout	The amount of time, in seconds, that passes before it is assumed the host did not receive the inform notification message. The range is 0 to 9999 and the default is 15.
Retry	The number of retries made when a response to a generated inform message is not received. The range is 0 to 255 and the default is 3.
Notify or Community	Indicates whether the SNMP host receives a trap or an inform notification message, or indicates the name of the community used to access the host ■ A trap is any message generated that contains unconfirmed PDUs. You may receive PDU traps for security models v1, v2c, and v3. ■ An inform is any message generated that contains confirmed PDUs. You may receive informs for only v2c and v3 security models.

Parameter	Description
Storage	Specifies how the host entry is stored. The options are: ■ volatile: The entry is stored in volatile memory. The information is lost during a system reboot. ■ nonvolatile (default): The entry is stored in non-volatile memory. The information is not lost during a system reboot. ■ permanent: The entry is stored in non-volatile memory. You cannot delete the information but you can make modifications. ■ readonly: The entry is stored in non-volatile memory. You cannot delete or modify the information.
Mask	The mask of the IP address on which the SNMP host is configured to receive traps.
MMS	The maximum message size (in bytes) of an SNMP message that the SNMP engine transmits or receives and processes. Values range from 484 to 65535. The default is 484.
Model	The SNMP security model used to process SNMP messages and gain access to the group. Your options are V1, V2c or V3.

Parameter	Description
Level	The level of security to process SNMP messages. You can choose one of the following three levels: ■ No Authentication: Provides no authentication and no encryption. This is the lowest level of security. V1 and V2c security models provide only this level of security. ■ Authentication: Provides authentication but no encryption. Only V3 security model provides this level of security. ■ Privacy: Provides authentication and encryption. This is the highest level of security. Only V3 security model provides this level of security.
Group Name	The community name associated to the SNMP group for the specific SNMP host.
Type	Indicates whether the notification is sent as a trap or inform.
Notifications Sent	The type of event for which you want to be notified. Notifications may be specified for system events occurring for clusters, modules, interfaces and DOCSIS modules.
Cluster events:	Cluster events refer to faults that affect the management module.
■ authentication-failure	SNMP receives a bad Community Name.
■ bcm-failover-down	Services are going down. This notification type applies to redundant configurations only.
■ bcm-failover-up	Services are coming up. This notification type applies to redundant configurations only.
■ bcm-state-change	A change in the craft port IP address.

Parameter	Description
■ bcm-sw-mismatch	The secondary will not come up because its software revision does not match the software revision of the primary.
■ trace-log	For ADC internal use only.
■ cold-start	Generated when module boots from power up.
■ warm-start	Generated when module boots from reset.
■ icl-state-change	A change in the ICL link.
Module- events:	Module events refer to hardware faults that affect any application module.
■ cable-modem-auth-failure	Cable modem failed authorization and did not register.
■ cable-modem-down	Cable modem is not operational.
■ cable-modem-up	Cable modem is operational.
■ card-down	Application module failure.
■ card-up	Application module is operational.
■ dhcp-relay-not-configured	DHCP configuration error.
■ local-sonet-alarm	A transmission problem is detected from the transmitter.
■ remote-sonet-alarm	A transmission problem is detected from the receiver.
■ Interface-related events:	Interface-related events refer to faults that affect the link state of the interface.
■ link-up	Link to IP network is operational.
■ link-down	Link to IP network is not operational.
■ chassis-fault	Auxiliary device-related event that refers to faults associated to the fan tray, power source and clock sources. *NOTE: Reference "Timing and Alarm Module Fault Reporting" on page 155, for a list of the chassis-related notification types*.

Parameter	Description
■ chassis-fault-cleared	Indicates the chassis event that caused a fault is fixed.
DOCSIS events:	DOCSIS events refer to initialization faults on DOCSIS and EuroDOCSIS modules.
■ docs-dyn-rsp-fail	A dynamic service response failure occurred during the dynamic services process.
■ docss-dyn-ack-fail	A dynamic service acknowledgement failure occurred during the dynamic services process.
■ docs-dyn-req-fail	A dynamic service request failure occurred during the dynamic services process.
■ docs-bpi-init	A BPI initialization attempt failure occurred during the registration process.
■ docs-bpkm	A baseline privacy key management operation failed.
■ docs-dcc-ack-fail	A dynamic channel change acknowledgement failed during the dynamic channel change process in the CMTS.
■ docs-dcc-req-fail	A dynamic channel change request failed during the dynamic channel change process in the cable modem and was detected by the CMTS.
■ docs-dcc-rsp-fail	A dynamic channel change response failed during the dynamic channel change process in the CMTS.
■ docs-dynamic-sa	A dynamic security association failed.
■ docs-init-ack-fail	A registration acknowledgement failure from the cable modem occurred during the cable modem initialization process and was detected on the CMTS side.
■ docs-init-req-fail	A registration request failure from the cable modem occurred during the cable modem initialization process and was detected on the CMTS side.

Parameter	Description
■ docs-init-rsp-fail	A registration response failure from the cable modem occurred during the cable modem initialization process and was detected on the CMTS side.
Provisioning events:	Provisioning events refer to faults that pertain to the FastFlow BPM running on the Cuda 12000.
■ duplicate-addr	A duplicate IP address has been detected.
■ isp-addr-high	The free IP address count exceeded the upper threshold for the specified ISP.
■ isp-addr-low	The free address count fell below the lower threshold for the specified ISP.
■ ldap-failed	A directory server access failure occurred.
■ ldap-restored	Directory server access is operational after a failure.
■ prov-service	A FastFlow BPM service started, stopped, or failed.
■ subnet-addr-high	The free IP address count exceeded the high available address threshold for a subnet.
■ subnet-addr-low	The free IP address count fell below the low available address threshold for a subnet.
Routing events:	Routing events refer to events that indicate a change in the state of OSPF neighbors and OSPF virtual neighbors.
■ ospf-nbr-state	Signifies a change in the state of an OSPF neighbor on a physical interface. To send this notification type, note that you also have to enable sending of OSPF neighbor state traps using the **report** command.

Parameter	Description
■ ospf-virt-nbr-state	Signifies a change in the state of an OSPF neighbor on a virtual interface. To send this notification type, note that you also have to enable sending of OSPF virtual neighbor state traps using the **report** command.
Modem deregistration event:	A modem deregistration event refers to the deregistration of cable modems.
■ dereg-modems	Signifies that a number or percentage of modems have deregistered over the deregistration time interval.

Perform the following tasks to configure event notifications. Refer to the configuration examples, below:

Tasks	Commands
1. Enter configuration mode.	**root**
2. Create an SNMP host to receive trap messages.	**snmp-server host** <ip address> <community-name> {**traps | informs** [**timeout** <seconds>] [**retries** <retries>]} [**version** {**1 | 2c | 3** {**auth | noauth | priv**}]
3. Specify the UDP port number on which the SNMP host will receive traps.	**snmp-server host** <ip address> <community-name> {**traps | informs [timeout** <seconds>] [**retries** <retries>]} [**version** {**1 | 2c | 3** {**auth | noauth | priv**}] **udp-port** <port>
4. Specify the maximum message size (MMS), in bytes, of an SNMP message that the SNMP engine will transmit or receive and process.	**snmp-server host** <ip address> <community-name> {**traps | informs** [**timeout** <seconds>] [**retries** <retries>]} [**version** {**1 | 2c | 3** {**auth | noauth | priv**}] [**mms**] <size>

Tasks	Commands
5. Specify the storage type for this host. By default, storage type is set to NonVolatile.	**snmp-server host** <ip address> <community-name> {**traps | informs** [**timeout** <seconds>] [**retries** <retries>]} [**version** {**1 | 2c | 3** {**auth | noauth | priv**}] [**storage** {**volatile | nonvolatile | permanent | readonly**}]
6. Specify the type of events for which you want to be notified.	**snmp-server host** <ip address> <community-name> {**traps | informs [timeout** <seconds>] [**retries** <retries>]} [**version** {**1 | 2c | 3** {**auth | noauth | priv**}] [**notification-type** <type>**...**]
7. Display SNMP host information.	**show snmp host** [**parameters**]
8. Display all SNMP hosts and associated notification type(s).	**show snmp notify** [<ip address>]
9. Remove an SNMP host entity from the trap recipient list, or remove a parameter in the host entry.	**no snmp-server host** <ip address> {**traps | informs**} [**mask**] [**notification-type** <type>**...**]

Example 1

The following example creates and displays an SNMP host with default parameter values:

```
root# snmp-server host 133.10.1.1 june informs retries 2  version 2c noauth
root# show snmp host

Host:Port           Time Retry Notify or      Storage     Mask            MMS
                    -out       Community
------------------- ---- ----- ------------- ----------- --------------- -----
133.10.1.1:162        15     2 inform        NonVolatile                   484

cli:192.168.208.3:root# show snmp host parameters

Notify:Host:Port             Model Level  Group Name                Storage
---------------------------- ----- ------ ------------------------- -----------
inform:133.10.1.1:162        V2c   NoAuth june                      NonVolatile

cli:192.168.208.3:root#
```

Example 2

The following example displays all SNMP hosts notification destinations and associated notification types.

```
root# show snmp notify

row count: 2

Host:Port            Storage     Notifications Sent                       Type
-------------------- ----------- ---------------------------------------- ------
136.4.6.6:164        NonVolatile trace-log cold-start link-up             inform
127.0.0.1:54321      NonVolatile prov-service ldap-failed ldap-restored   V2
                                 subnet-addr-low subnet-addr-high
                                 isp-addr-low isp-addr-high
                                 duplicate-addr bcm-failover-down
                                 bcm-failover-up bcm-sw-mismatch
                                 card-down card-up trace-log
                                 cable-modem-up cable-modem-down
                                 bcm-state-change icl-state-change
                                 cable-modem-auth-failure
                                 dhcp-relay-not-configured
                                 local-sonet-alarm remote-sonet-alarm
                                 chassis-fault chassis-fault-cleared
                                 cold-start warm-start link-down link-up
                                 authentication-failure
```

Monitoring SNMP

The **show snmp** command allows you to monitor SNMP activity on the Cuda 12000. To use this command, perform the following tasks:

Tasks	Commands
1. Enter root mode.	**root**
2. Monitor SNMP activity.	**show snmp**

The command displays the following information:

Table 9-9 SNMP Parameters and Statistics

Parameter	Description
Contact	The type of contact for this network. The contact is typically a network administrator's name, extension, and/or e-mail address.
Name	The name of the system (sysName MIB object).
Location	The physical location of the device (sysLocation MIB object).
SNMP Packets Received	Total number of messages that the transport service delivers to the SNMP entity.
Bad SNMP Version Errors	Total number of SNMP messages that the SNMP entity receives using an unsupported version of SNMP.
Unknown Community Names	Total number of SNMP messages that the SNMP entity receives using an SNMP community name not known to the entity.
Illegal Community Names	Total number of SNMP messages that the SNMP entity receives that represent an SNMP operation that is now allowed by the SNMP community named in the message.
Encoding Errors	Total number of ASN.1 or BER errors that the SNMP entity encounters when decoding SNMP messages.
Silent Drops	Total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs, and InformRequest-PDU packets that the SNMP entity receives and drops.
Unknown Security Models	Total number of packets that the SNMP engine receives and drops because the security model was not known or supported by the SNMP engine.

Parameter	Description
Invalid Messages	Total number of packets that the SNMP engine receives and drops because there were invalid or inconsistent components in the SNMP message.
Unknown PDU Handlers	Total number of packets that the SNMP engine receives and drops because the PDU contained in the packets could not be passed to an application responsible for the PDU type.
Authentication Traps	Indicates whether the SNMP entity is able to generate failure traps.

Example

In the following example, the user issues the show snmp command to monitor SNMP activity.

```
root# show snmp
Contact                    router
Name                       cuda 111
Location                   bldg. 1400
SNMP packets received                182168
Bad SNMP version errors                   0
Unknown community names                   0
Illegal community names                   0
Encoding errors                           0
Silent drops                              0
Unknown security models                   0
Invalid messages                          0
Unknown PDU handlers                      0
Authentication traps                 disable

cli:192.168.208.3:root#
```

Sample SNMP Configurations

This section provides sample configurations for SNMPv1/v2c community access control, SNMPv3 access control, and notification.

Sample SNMPv1/v2c Community Access Control

To configure SNMPv1/v2c community access control, you must:

1. Configure SNMP Access Views.
2. Configure SNMP Groups.
3. Configure SNMPv1, v2c Communities.

In this sample configuration, the administrator creates three communities (and associated views and groups):

- A community called "monitor" that allows any host read-only access to the entire MIB, except for sensitive SNMP configuration information. No write access is allowed.
- A community called "admincon" that allows read-write access to the entire MIB, but only from management hosts in a particular address range (such as a management network operations center). In this case, the address range is 100.100.0.0 through 100.100.255.255.
- A community called "justme" that allows the same access as the "admincon" community, but from two individual hosts only.

To configure the "monitor" community, the administrator first issues the following commands to configure two read-only views, each named "nosnmpconfig:"

```
cli:192.168.208.3:root# snmp-server view nosnmpconfig 1.3.6.1 included
cli:192.168.208.3:root# snmp-server view nosnmpconfig snmpModules excluded
```

The administrator then creates two groups named "monitorgroup" that associate the read-only view (nosnmpconfig) and the community "monitor," which is created afterward.

```
cli:192.168.208.3:root# snmp-server group monitorgroup v1 read nosnmpconfig
cli:192.168.208.3:root# snmp-server group monitorgroup v2 read nosnmpconfig
```

The administrator then creates the community "monitor," which includes an association to the group named "monitorgroup."

```
cli:192.168.208.3:root# snmp-server community monitor monitorgroup
```

To configure the "admincon" community, the administrator issues the following commands:

```
cli:192.168.208.3:root# snmp-server view allaccess 1.3.6.1 included
cli:192.168.208.3:root# snmp-server group admingroup v1 read allaccess write
allaccess
cli:192.168.208.3:root# snmp-server group admingroup v2 read allaccess write
allaccess
cli:192.168.208.3:root# snmp-server community admincon admingroup address
100.100.0.0 mask 255.255.0.0
```

To configure the "justme" community, the administrator issues the following commands:

```
cli:192.168.208.3:root# snmp-server community justme admingroup address 100.100.10.5
cli:192.168.208.3:root# snmp-server community justme admingroup address 100.100.10.8
```

Notice that the administrator does not have to specify a view or a group. The administrator uses the view and group created during configuration of the "admincon" community.

Sample SNMPv3 Access Control

To configure SNMPv3 community access control, you must:

1. Configure SNMP Access Views.
2. Configure SNMP Groups.
3. Configure SNMPv3 Users and Contexts.

First, the administrator creates:

- A view that includes access to most of the MIB and a view that excludes access from sensitive configuration information.
- A group that configures the user for the default security model "noauth."
- An SNMPv3 user called "mgr."
- A context called "monitor" that allows the user read-only access to the entire MIB, except for sensitive SNMP configuration information.

To configure these access control elements, the administrator issues the following commands:

```
cli:192.168.208.3:root# snmp-server view nosnmpconfig 1.3.6.1 included
cli:192.168.208.3:root# snmp-server view nosnmpconfig snmpModules excluded
cli:192.168.208.3:root# snmp-server group mgr v3 noauth read nosnmpconfig context
monitor
cli:192.168.208.3:root# snmp-server context monitor
cli:192.168.208.3:root# snmp-server user mgr
```

The administrator then decides that a user with read and write access to the entire MIB is needed. To create this user and assign the user the necessary access privileges, the administrator issues the following commands:

```
cli:192.168.208.3:root# snmp-server view allaccess 1.3.6.1 included
cli:192.168.208.3:root# snmp-server group superman v3 priv read allaccess write
allaccess context admin
cli:192.168.208.3:root# snmp-server context admin
cli:192.168.208.3:root# snmp-server user superman auth md5 ab03045f6e priv des56
a0b0c0d0e0f0
```

The group entry allows the context "admin" to have read and write access to the entire MIB. Only management hosts that use the user "superman" and the context "admin" can access the view "allaccess."

Sample Notification Configuration

The following sample commands configure the Cuda 12000 to send SNMPv1 traps to a host (201.1.1.20):

```
cli:192.168.208.3:root# snmp-server view allaccess 1.3.6.1 included
cli:192.168.208.3:root# snmp-server group trapcommunity v1 notify allaccess
cli:192.168.208.3:root# snmp-server group trapcommunity v2 notify allaccess
cli:192.168.208.3:root# snmp-server community trapcommunity trapcommunity
cli:192.168.208.3:root# snmp-server host 201.1.1.20 trapcommunity traps version 1
```

In this example, the SNMP agent on the Cuda 12000 sends SNMPv1 traps (on the default UDP port of 162) to a host with an IP address of 201.1.1.20. Because the administrator does not specify any notification types, all types are sent.

The administrator creates two group entries for community-based SNMPv1/v2c access. Each group entry is assigned a notify view of "allaccess" that allows notifications access to the entire MIB (this community has no read or write access). The community inserted into outgoing traps will be "trapcommunity."

To send SNMPv2 traps instead of SNMPv1 traps, the administrator would issue the same commands except for a slightly different version of the **snmp-server host** command:

```
cli:192.168.208.3:root# snmp-server host 201.1.1.20 trapcommunity traps version 2c
```

To send inform messages instead of traps, another form of the **snmp-server host** command would be used:

```
cli:192.168.208.3:root# snmp-server host 201.1.1.20 trapcommunity informs version 2c
```

This command sends inform messages with the default timeout and retries values set. To change the defaults to 20 (for timeout) and 5 (for retries), the administrator would issue the following command:

```
cli:192.168.208.3:root# snmp-server host 201.1.1.20 trapcommunity informs timeout 20
retries 5 version 2c
```

10 MANAGING SYSTEM EVENTS

This chapter describes how to manage event transmission and includes the following sections:

About System Events

An event is a problem, a configuration change or some other noteworthy incident that occurs on the Cuda 12000 or in the network. Events create the generation of:

- System log (syslog) messages
- SNMP traps, which the Cuda 12000 sends to network management stations
- Internal log messages

Configuring the Syslog Server

Before you manage event transmission or reporting using the syslog server, you set the IP address of the syslog server to which your Cuda 12000 writes system log messages, as required by DOCSIS 1.1 standards.

You may specify the IP address of the local Syslog server on your Cuda 12000 or a remote syslog server on another Cuda 12000.

To configure the IP address of the Syslog server, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Specify the IP address of the syslog server.	**event-config syslog** <ip-address>
3. Display the syslog server IP address. If a Syslog server IP address does not currently exist, the default is 0.0.0.0.	**show event-config syslog**

Example

```
cli:192.168.208.3:root# show event-config syslog
Syslog Server                          0.0.0.0

cli:192.168.208.3:root#
```

To remove the Syslog server entry, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Remove the syslog server entry.	**event-config syslog 0.0.0.0**

Configuring SNMP Trap Recipients

You must define a list of IP addresses of SNMP management stations that receive traps or syslog messages from your Cuda 12000. Use this procedure to specify each trap recipient:

Task	Command
1. Enter root mode.	**root**
2. Specify the trap recipient.	**snmp-server host** <ip-address> [**traps** | **informs** [**timeout** <seconds>][**retries** <retries>]] [**version** {**1** | **2c** | **3**} {**auth** | **noauth** | **priv**}] <community-name> [**udp-port** <port>] [**mms** <size>] [**storage** {**volatile** | **nonvolatile** | **permanent** | **readonly**}] [**notification-type** <type>...] *For information on command arguments, refer to Chapter 9, Simple Network Management Protocol (SNMP), on page page 161, or see **snmp-server host** command in the Cuda 12000 IP Access Switch CLI Reference Guide.*
3. Display the trap recipient.	**show snmp host** <ip-address>

Example

```
root# snmp-server host 136.4.6.6 informs timeout 500 retries 200 version 2c noauth
private udp-port 164 mms 5000 notification-type cold-start link-up
cli:192.168.220.230:root#
```

Removing SNMP Trap Recipients

Perform this task to remove an SNMP trap recipient:

Task	Command
Remove the trap recipient.	**no snmp-server host** <ip-address>

Example

```
root# no snmp-server host 136.4.6.6
```

Configuring Event Transmission

A Cuda 12000 can generate a significant volume of events in a short period of time. The Cuda 12000 manages event transmission in compliance with DOCSIS 1.1 standards.

To avoid flooding the syslog server and network management stations with events, you can control the pace of event transmission by configuring these parameters:

Table 10-1 Event Transmission Parameters

Parameter	Description
Event Threshold	Number of events that the Cuda 12000 may generate per event interval before throttling occurs. Throttling is the process of eliminating excessive events. Note that an event causing both a trap and a syslog message is still treated as a single event. Values range from 0 to 4294967295. The default is 0.
Event Interval	The interval, in seconds, over which the event threshold applies. For example, if you configure an event threshold of 20 and an event interval of 40 seconds, then the Cuda 12000 may generate 20 events over 40 seconds before throttling occurs. Values range from 0 seconds to 2147483647 seconds. The default is 1.
Event Administrative Status	Controls the transmission of traps and syslog messages with respect to the event threshold. Specify one of these administrative status values: ■ unconstrained (default) — The Cuda 12000 transmits traps and syslog messages without regard to the event threshold and interval settings. ■ maintainBelowThreshold — The Cuda 12000 suppresses traps and syslog messages if the number of events exceeds the threshold. The Cuda 12000 resumes transmitting traps and syslog messages when the number of events drops below the threshold. ■ stopAtThreshold — The Cuda 12000 stops trap and syslog message transmissions at the threshold. To resume trap and syslog message transmission, you must reset the threshold. ■ inhibited – The Cuda 12000 suppresses all trap transmissions and syslog messages.

Parameter	Description
Throttle Inhibited	Displays the throttle inhibited status. This field displays True if one of the following conditions is met: ■ Event Administrative Status is set to inhibited. ■ Event Administrative Status is set to stopAtThreshold and the threshold has been reached. Otherwise, this field displays False.

To configure event transmission, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Specify the event threshold.	**event-config throttle threshold** <number>
3. Specify the event threshold interval.	**event-config throttle interval** <number>
4. Specify the event administrative status.	**event-config throttle admin** {**unconstrained** | **maintainBelowThreshold** | **stopAtThreshold** | **inhibited**}
5. Display the event threshold.	**show event-config throttle**

Example

The following is an example of an event threshold configuration, using the default settings:

```
cli:192.168.208.3:root# show event-config throttle
Event Throttle Parameters
-------------------------
Threshold                                0
Interval                                 1
Admin Status                    unconstrained
Throttle Inhibited                      False

cli:192.168.208.3:root#
```

Event Reporting

Each Cuda 12000 event belongs to one of eight event classes. An event class defines the severity of the event. You can configure each event class to be sent through a subset of reporting mechanisms (trap, syslog, or local event log). To do this, you specify:

- An event class
- How you want events in that class to be reported

Event Classes

Event classes are ordered from most critical (emergency) to least critical (debug). The following table lists the event classes, in priority order:

Table 10-2 Event Classes

Event Class	Description
Emergency	Indicates hardware- or software-related problems with DOCSIS or EuroDOCSIS modules. Prevents CMTS operation.
Alert	Indicates a serious failure that causes the Cuda 12000 to reboot.
Critical	Indicates a serious failure that requires attention and prevents the device from transmitting data. Failure may be resolved without a system reboot.
Error	Indicates a failure occurred that could interrupt the normal data flow.
Warning	Indicates a failure occurred that could interrupt the normal data flow. (This failure is not as severe as reported for Error events.)
Notice	Indicates an event that requires attention, but is not a failure.
Information	Indicates an event that may be helpful for tracing normal operation. Informational events do not report failures.
Debug	An event used for only debugging purposes.

Reporting Actions

Each event class is associated with a reporting action. The following table lists the reporting actions:

Table 10-3 Reporting Actions

Reporting Action	Description
local	Write a message to the internal log.
local|traps	Write a message to the internal log and send a trap.
local|syslog	Write a message to the internal log and send a syslog message.
local|traps|syslog	Write a message to the internal log, send a trap, and send a syslog message.
none	Do not report events in this class.

Configuring Event Reporting

By default, the Cuda 12000 reports events as follows:

Table 10-4 Default Event Class Reporting Actions

Event Class	Default Reporting Action
Emergency	local
Alert	local
Critical	local|traps|syslog
Error	local|traps|syslog
Warning	local|traps|syslog
Notice	local|traps|syslog
Information	none
Debug	none

To configure event classes and associated reporting actions, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Assign the Default event class and reporting action.	**event-config reporting default**
3. Assign the Emergency event class and associated reporting action.	**event-config reporting emergency local**
4. Assign the Alert event class and associated reporting action.	**event-config reporting alert local**
5. Assign the Critical event class and associated reporting action. *Note: The pipe (|) must be included in the command string.*	**event-config reporting critical** {**local|traps|syslog**} {**local|traps}** {**local|syslog**}
6. Assign the Error event class and associated reporting action. *Note: The pipe (|) must be included in the command string.*	**event-config reporting error** {**local|traps|syslog**} {**local|traps**} {**local|syslog**}
7. Assign the Warning event class and associated reporting action. *Note: The pipe (|) must be included in the command string.*	**event-config reporting warning** {**local|traps|syslog**} {**local|traps**} {**local|syslog**}
8. Assign the Notice event class and associated reporting action. *Note: The pipe (|) must be included in the command string.*	**event-config reporting notice** {**local|traps|syslog**} {**local|traps**} {**local|syslog**}
9. Assign the Information event class and associated reporting action.	**event-config reporting info**
10. Assign the Debug event class and associated reporting action.	**event-config reporting debug**

Refer to the next section for information on viewing the event reporting configuration.

Viewing Event Reporting Configuration

You may view the event reporting configuration. The output includes event reporting configuration for all current event classes.

To view the current event reporting configuration, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Display current event reporting configuration	**show event-config reporting**

Example

The following example displays the current event reporting configuration.

```
cli:192.168.208.3:root# show event-config reporting
Event Reporting Priorities
--------------------------

row count: 8

Priority    Action
----------- ------------------
  emergency local
      alert local
   critical local|syslog
      error local|traps
    warning local|traps|syslog
     notice local|traps|syslog
information none
      debug none

cli:192.168.208.3:root#
```

Event Classes and SNMP System Events

Event classes are associated with SNMP system events, as shown in Table 10-5. For additional information about SNMP System Events refer to Chapter 9, *Simple Network Management Protocol (SNMP),* on page 161.

Table 10-5 List of System Events and Their Event Classes

SNMP System Event	Event Class
Cluster events:	
■ authentication-failure	Notice
■ bcm-failover-down	Notice
■ bcm-failover-up	Notice
■ bcm-state-change	Notice
■ bcm-sw-mismatch	Notice
■ trace-log	Notice
■ cold-start	Notice
■ warm-start	Notice
■ icl-state-change	Notice
Module events:	
■ cable-modem-auth-failure	Notice
■ cable-modem-down	Critical
■ cable-modem-up	Notice
■ card-down	Critical
■ card-up	Notice
■ dhcp-relay-not-configured	Warning
■ local-sonet-alarm	Error
■ remote-sonet-alarm	Error
■ Interface-related events:	Notice
■ link up	Notice
■ link down	Error
■ chassis-fault	Critical
■ chassis-fault-cleared	Notice

SNMP System Event	Event Class
Provisioning events:	
■ duplicate-addr	Notice
■ isp-addr-high	Notice
■ isp-addr-low	Notice
■ ldap-failed	Notice
■ ldap-restored	Notice
■ prov-service	Notice
■ subnet-addr-high	Notice
■ subnet-addr-low	Notice
DOCSIS events:	
■ docs-dyn-rsp-fail	Warning
■ docss-dyn-ack-fail	Warning
■ docs-dyn-req-fail	Warning
■ docs-bpi-init	Informational
■ docs-bpkm	Error
■ docs-dcc-ack-fail	Warning
■ docs-dcc-req-fail	Warning
■ docs-dcc-rsp-fail	Warning
■ docs-dynamic-sa	Warning
■ docs-init-ack-fail	Warning
■ docs-init-req-fail	Warning
■ docs-init-rsp-fail	Warning
Routing events:	
■ ospf-nbr-state	Notice
■ ospf-virt-nbr-state	Notice
Modem deregistration event:	
■ dereg-modems	Warning

Clearing the Event Log

To prevent your internal event log from consuming too much disk space, you may want to clear the log periodically. Use this procedure to clear the event log:

Task	Command
1. Enter root mode.	**root**
2. Clear the event log.	**event-log clear**

Displaying Event Transmission, Reporting, and Syslog Parameters

Use this procedure to display the event transmission, reporting, and syslog parameters:

Task	Command
1. Enter root mode.	**root**
2. Display event transmission, reporting, and syslog parameters.	**show event-config** {**throttle** | **reporting** | **syslog**}

Example

```
root# show event-config
Event Throttle Parameters
-------------------------
Threshold                                    0
Interval                                     1
Admin Status                       unconstrained
Throttle Inhibited                         True

Event Reporting Priorities
--------------------------

row count: 8

Priority    Action
----------- ------------------
  emergency local
      alert local
   critical local|traps|syslog
      error local|traps|syslog
    warning traps|syslog
     notice traps|syslog
information none
      debug none

Syslog Server                      133.132.1.1

cli:192.168.208.3:root# show event-config throttle
Event Throttle Parameters
-------------------------
Threshold                                    0
Interval                                     1
Admin Status                       unconstrained
Throttle Inhibited                         True

cli:192.168.208.3:root# show event-config reporting
Event Reporting Priorities
--------------------------

row count: 8

Priority    Action
----------- ------------------
  emergency local
      alert local
   critical local|traps|syslog
      error local|traps|syslog
```

```
    warning traps|syslog
     notice traps|syslog
information none
      debug none

cli:192.168.208.3:root# show event-config syslog
Syslog Server                    133.132.1.1

cli:192.168.208.3:root#
```

Displaying the Event Log

Use this procedure to display the log of events that the Cuda has generated:

Task	Command
1. Enter root mode.	**root**
2. Display the contents of the event log.	**show event-log**

The **show event-log** command output displays these fields of information about each event:

Table 10-6 Event Log Fields

Field	Description
Index	The number of the event in the log. This number is used to order the events in the log.
First Time	The time that the log entry was created.
Last Time	The time that the last event associated with the log entry occurred. In some cases, multiple events can be associated with a single log entry. This tends to happen when duplicate events are reported. However, when only one event is reported, then one event is associated with an entry, which means that the First Time and Last Time values are the same.
Counts	The number of consecutive event instances that this event entry reports. The count starts at 1 when the entry is created and increments by one for each subsequent duplicate event.
Level	The event's class (emergency, alert, critical, error, warning, notice, info, debug).

Field	Description
ID	An internal event identifier. The Text field describes the event associated with this identifier.
Text	Brief description of the event.

Example

```
root# show event-log

row count: 133

Index  First Time Last Time  Counts     Level      ID         Text
------ ---------- ---------- ---------- ---------- ---------- ----------
     1 2000-12-31 2000-12-31          1   critical 2147483652 CMTS/CM
       ,21:1:40.0 ,21:1:40.0                                  Down -
       ,455:0     ,455:0                                      ifIndex =
                                                              8781825
     2 2000-12-31 2000-12-31          1   critical 2147483649 Card Down
       ,21:31:40. ,21:31:40.                                  - 1/1/1
       0,455:0    0,455:0
     3 2001-1-1,1 2001-3-6,1       1264   critical 2147483652 CMTS/CM
       :20:0.0,45 :26:40.0,4                                  Down -
       5:0        55:0                                        ifIndex =
                                                              8781825
     4 2000-12-31 2000-12-31          2   critical 2147483652 CMTS/CM
       ,19:28:20. ,19:28:20.                                  Down -
       0,455:0    0,455:0                                     ifIndex =
                                                              8781825
     5 2000-12-31 2000-12-31          5   critical 2147483652 CMTS/CM
       ,19:36:40. ,23:46:40.                                  Down -
       0,455:0    0,455:0                                     ifIndex =
--More--
```

III IP ROUTING

11 CREATING ROUTE FILTERS

This chapter provides information and procedures on how to create route filters to control the flow of routes on your network. You create these route filters in the form of route-maps and map-lists. Route-maps contain the fundamental gating action *(*permit or deny*)* based on selected route-match criteria with optional override actions. Map-lists are sequential groupings of these route-maps.

This chapter includes the following sections:

About RIP and OSPF Route Maps

The system uses route filtering functions to control the flow of routes to and from other RIP and OSPF routers. Two filtering functions are supported for control of RIP and OSPF routes:

- **Import** — Controls how routes are added to the system's routing table.
- **Export** — Controls which routes are advertised to other routers.

Route filtering is used to customize connectivity, increase security, conserve routing table space, or adjust route cost.

In order to understand route filtering, you must be familiar with the following functions:

- **Route-Map** — The route map defines the match criteria and the action that you want the system to take when a match is found. You use the **route-map** command to create a route-map and enter configuration mode for the route-map. When you define a route-map, you specify a permit or deny action. You then define the match criteria of a route map using the **match** command. In addition, you can choose to define override actions using the **override** command.
- **Map-List** — A sequential grouping of route-maps. Routes are sequentially compared against all route-maps that comprise the map-list. When a match is found, the action defined by the route-map is invoked, and further exploration of the route-map is ended. If the system does not find a match, then no action is taken.

Defining RIP and OSPF route filtering is a two-step process:

- First, you create route maps to define match criteria and the action that you want the system to take when it finds a match. You can choose to permit or deny a route, as well as override route cost, tag, or preference.
- Second, you arrange those route-maps into map-lists.

When defining route maps and map lists, remember the following:

- Map lists are made up of one or more route maps.
- The same route map may be shared among multiple map lists.
- Route maps within each map list must be sorted in order of the specific-to-general match criteria and action needs of the map list.

Creating Route Maps

You can use route maps to control and modify routing information and to define the conditions by which routes are redistributed.

When you run the **route-map** command within router:rip:import or export mode, or router:ospf:export mode the following syntax applies:

route-map <map-tag> {**permit** | **deny**}

The map-tag is a number that identifies the route map. The **permit** keyword configures the router to accept matching routes; the **deny** keyword configures the router to reject matching routes.

When you run the **route-map** command within router-ospf import mode, the following syntax applies:

route-map <map-tag>

All route maps must permit incoming OSPF routes; **deny** is not an option. Once you issue the **route-map** command, you enter configuration mode for the specified route-map. For example:

```
cli:172.16.19.10:root# router ospf export
mode: router:ospf:export
cli:172.16.19.10:router:ospf:export# route-map 80 permit
cli:172.16.19.10:router:ospf:export:route-map(80)#
```

While within this mode, you then use the following commands:

- **match** — Use this command to specify the routes to match.
- **override** — Use this command to alter the attributes of redistributed routes.

While within any route-map mode, you can display a summary of all route maps configured within that mode using the **show route-map** command as follows:

```
cli:172.16.19.10:router:rip:export:route-map(3)# show route-map

row count: 3

ID  Description Route Address   Route Mask      Interface Address Owner
--- ----------- --------------- --------------- ----------------- -----
  1             172.16.0.0      255.255.0.0     0.0.0.0            NONE
  2             10.255.0.0      255.255.0.0     0.0.0.0            NONE
  3             10.0.0.0        255.0.0.0       0.0.0.0            NONE

cli:172.16.19.10:router:rip:export:route-map(3)#
```

Using the Match Command

Use the **match** command within route-map configuration mode to define the match criteria for the route map. Refer to the following sections for more information on using this command:

- "Creating OSPF Import Route Maps" on page 229
- "Creating OSPF Export Route Maps" on page 231
- "Creating RIP Import Route Maps" on page 234
- "Creating RIP Export Route Maps" on page 236

The following example sets match attributes for route-map 80, then verifies the new route-map configuration using the **show route-map** command:

```
cli:172.16.19.10:router:ospf:export:route-map(80)# match ip-address 172.16.19.0
255.255.255.0
cli:172.16.19.10:router:ospf:export:route-map(80)# match route-type rip
cli:172.16.19.10:router:ospf:export:route-map(80)# show route-map 80
ID                                      80
Description
Route Address                    172.16.19.0
Route Mask                     255.255.255.0
Type                                  RIP
Specific1                         0.0.0.0
Specific2                         0.0.0.0
Tag                                     0
Key Bits                                5
Metric                                  0
Flags                                   1
Action Tag                              0

cli:172.16.19.10:router:ospf:export:route-map(80)#
```

Using the Override Command

Use the **override** command within route-map configuration mode to specify override actions to take for all matching routes. Refer to the following sections for more information on using this command:

- "Creating OSPF Import Route Maps" on page 229
- "Creating OSPF Export Route Maps" on page 231
- "Creating RIP Import Route Maps" on page 234
- "Creating RIP Export Route Maps" on page 236

The following example configures route-map 80 to redistribute matching routes with a cost metric of 10.

```
cli:172.16.19.10:router:ospf:export:route-map(80)# override metric 10
cli:172.16.19.10:router:ospf:export:route-map(80)# show route-map 80
ID                                        80
Description
Route Address                    172.16.19.0
Route Mask                     255.255.255.0
Type                                    RIP
Specific1                         0.0.0.0
Specific2                         0.0.0.0
Tag                                        0
Key Bits                                   5
Metric                                    10
Flags                                     17
Action Tag                                 0

cli:172.16.19.10:router:ospf:export:route-map(80)#
```

Creating OSPF Import Route Maps

You can use OSPF import route-maps to override the preference of incoming OSPF routes. Preference is the local ranking of the route.

OSPF import route maps are created within router-ospf import mode using the **route-map** command. To create an OSPF import route map, use this procedure:

Permit and deny options do not apply to OSPF import routes as all OSPF routes are always learned.

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Enter import mode.	**import**
3. Specify the route map or create a new one if it does not already exist.	**route-map** <map-tag>
4. Define that match criteria for this route map.	**match** {**ip-address** <ip-address> <mask> | **neighbor** <ip-address> <mask> | **tag** <tag-value> {**exact** | **exclude**}}
5. Define the override criteria that you want the system to apply to any routes that match.	**override preference** <preference-value>
6. *Optional:* You can then verify the route map with the **show route-map** command.	**show route-map** <map-tag>

The following example creates an OSPF import route map that assigns a preference of 100 to all incoming routes from the 172.16.0.0 network:

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# import
mode: router:ospf:import
cli:172.16.19.10:router:ospf:import# route-map 1
cli:172.16.19.10:router:ospf:import:route-map(1)# match ip-address 172.16.0.0
255.255.0.0
cli:172.16.19.10:router:ospf:import:route-map(1)# override preference 100
cli:172.16.19.10:router:ospf:import:route-map(1)# show route-map 1
ID                                      1
Description
Route Address                   172.16.0.0
Route Mask                    255.255.0.0
Peer Address                      0.0.0.0
Peer Mask                         0.0.0.0
Tag                                     0
Key Bits                                1
Preference                            100
Flags                                2049

cli:172.16.19.10:router:ospf:import:route-map(1)#
```

Creating OSPF Export Route Maps

You can use OSPF route maps to permit or deny advertisement of routes learned from a non-OSPF protocol. For example, you can choose to advertise select routes onto your OSPF network if they were originally learned through the RIP protocol, or if they were manually added as a static route. You can also override the cost metric of incoming routes originating for a non-OSPF protocol. When there are multiple routes to the same destination, the route with the lowest cost is preferred.

OSPF export route maps are created within router-ospf export mode using the **route-map** command. To create an OSPF export route map, use this procedure:

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Enter export mode.	**export**
3. Specify the route map that you want to configure, or create a new one if it does not already exist.	**route-map** <map-tag> {**permit** | **deny**}

Task	Command
4. Define the match criteria for this route map.	**match** {**ip-address** <ip-address> <mask> | **tag** <tag-value> {**exact** | **exclude**} | **specific1** <specific1-value> | **specific2** <specific2-value> | **route-type** {**none** | **connected** | **static** | **special** | **rip** | **bgp-ext** | **bgp-int** | } *Note that a "connected" route is a local route (a route to a directly connected network).*
5. Define the override criteria that you want the system to apply to any routes that match.	**override** {**metric** <metric-value> | **tag** <tag-value>}
6. *Optional:* You can then verify the route map with the **show route-map** command.	**show route-map** <map-tag>

The following example creates an export route map that prevents the 176.16.0.0 RIP network from being advertised.

```
cli:172.16.19.10:router:ospf:import:route-map(1)# router rip
mode: router:rip
cli:172.16.19.10:router:rip# export
mode: router:rip:export
cli:172.16.19.10:router:rip:export# route-map 1 deny
cli:172.16.19.10:router:rip:export:route-map(1)# match ip-address 172.16.0.0 255.255.0.0
cli:172.16.19.10:router:rip:export:route-map(1)# show route-map 1
ID                                      1
Description
Route Address                   172.16.0.0
Route Mask                    255.255.0.0
Interface Address                 0.0.0.0
Owner                                NONE
Specific                          0.0.0.0
Peer Mask                         0.0.0.0
Tag                                     0
Key Bits                                1
Metric                                 16
Flags                                   0
Action Tag                              0

cli:172.16.19.10:router:rip:export:route-map(1)#
```

Creating RIP Import Route Maps

You can use RIP import route maps to alter the preference of incoming RIP routes. When there are multiple routes to the same destination, the route with the numerically highest preference is preferred.

RIP import route maps are created within router-rip import mode using the **route-map** command. To create a RIP import route map, use this procedure:

Task	Command
1. Enter router-rip mode.	**router rip**
2. Enter import mode.	**import**
3. Specify the route map or create a new one if it does not already exist.	**route-map** <map-tag>
4. Define that match criteria for this route map.	**match** {**ip-address** <ip-address> <mask> | **tag** <tag-value> {**exact** | **exclude**} | **peer-address** <ip-address> <mask>}
5. Define the override criteria that you want the system to apply to any routes that match.	**override** {**metric** <metric-value> | **tag** <tag-value> | **preference** <preference-value>}
6. *Optional:* You can then verify the route map with the **show route-map** command.	**show route-map** <map-tag>

The following example creates a RIP import route-map that prevents the 176.16.0.0 network learned through a non-RIP protocol from being advertised via the RIP protocol.

```
cli:172.16.19.10:root# router rip
mode: router:rip
cli:172.16.19.10:router:rip# import
mode: router:rip:import
cli:172.16.19.10:router:rip:import# route-map 1 deny
cli:172.16.19.10:router:rip:import:route-map(1)# match ip-address 172.16.0.0 255.255.0.0
cli:172.16.19.10:router:rip:import:route-map(1)# show route-map 1
ID                                      1
Description
Route Address                    172.16.0.0
Route Mask                     255.255.0.0
Peer Address                       0.0.0.0
Peer Mask                          0.0.0.0
Tag                                     0
Key Bits                                1
Preference                              0
Metric                                  0
Flags                                   0
Action Tag                              0

cli:172.16.19.10:router:rip:import:route-map(1)#
```

Creating RIP Export Route Maps

You can use RIP export route maps to permit or deny advertisement of routes learned from non-RIP protocols. For example, you can choose not to advertise select routes via RIP if they were manually added as static routes.

You can also choose to override the cost metric of advertised routes originating for a non-RIP protocol. When there are multiple routes to the same destination, the route with the lowest cost metric is preferred.

OSPF export route maps are created within router-rip export mode using the **route-map** command. To create a RIP export route map, use this procedure:

Task	Command
1. Enter router-rip mode.	**router rip**
2. Enter export mode.	**export**
3. Specify the route map or create a new one if it does not already exist.	**route-map** <map-tag> {**permit** | **deny**}

Task	Command
4. Define that match criteria for this route map.	**match** {**ip-address** <ip-address> <mask> | **tag** <tag-value> {**exact** | **exclude**} | **interface-address** <ip-address> | **peer-mask** <mask> | **specific** <specific-value> | **route-owner** {**none** | **connected** | **ospf** | **ospf-ext** | **static** | **special** | **bgp-ext** | **bgp-int** | } *Note that a "connected" route is a local route (a route to a directly connected network).*
5. Define the override criteria that you want the system to apply to any routes that match.	**override** {**metric** <metric-value> **| tag** <tag-value>}
6. *Optional:* You can then verify the route map with the **show route-map** command.	**show route-map** <map-tag>

The following example creates a RIP export route map that allows the 176.16.0.0 network that was learned through a non-RIP protocol to be advertised via RIP with a cost of 16.

```
cli:172.16.19.10:root# router rip
mode: router:rip
cli:172.16.19.10:router:rip# export
mode: router:rip:export
cli:172.16.19.10:router:rip:export# route-map 1 permit
cli:172.16.19.10:router:rip:export:route-map(1)# match ip-address 172.16.0.0 255.255.0.0
cli:172.16.19.10:router:rip:export:route-map(1)# override metric 16
cli:172.16.19.10:router:rip:export:route-map(1)# show route-map 1
ID                                      1
Description
Route Address                   172.16.0.0
Route Mask                    255.255.0.0
Interface Address                 0.0.0.0
Owner                                NONE
Specific                          0.0.0.0
Peer Mask                         0.0.0.0
Tag                                     0
Key Bits                                1
Metric                                 16
Flags                                  17
Action Tag                              0

cli:172.16.19.10:router:rip:export:route-map(1)#
```

Creating Map Lists

A map-list is a sequential grouping of route maps. These route-maps serve as the filter criteria within the map-list. A route is sequentially compared against all route maps that comprise the *active* route list. Upon finding a match, the system takes the action defined by the route map and exists the list.

You create a map-list by adding a route map to it using the **map-list** command. You can create multiple map-lists, but only one list can be configured as the active list for each of the following areas:

- RIP Imports
- RIP Exports
- OSPF Imports
- OSPF Exports

Use the **set active** option to define a map-list as active. This means, if you create 3 map lists to define RIP import policies, only a single map list can be designated as the active map list against which incoming RIP routes are applied.

You add one or more route-maps to a specified map-list using the **map-list** command import or export modes. The following syntax applies:

map-list <route-map-list-number> **route-map** <route-map-number>

For example, the following command adds route-map 10 to map-list 1:

```
cli# map-list 1 route-map 10
```

For example, the following example creates map-list 1 then configures it as as the active map-list to which incoming RIP routes are applied:

```
cli:172.16.19.10:root# router rip import
mode: router:rip:import
cli:172.16.19.10:router:rip:import# map-list 1 route-map 1
cli:172.16.19.10:router:rip:import# map-list 1 set active
cli:172.16.19.10:router:rip:import#
```

Route-maps are appended to the specified map-list. The order in which you add the route-maps to the map-list determine the sequence in which the system examines the route maps; the first route-map that you added to the list is examined first, the final route-map that you appended to the list is examined last.

You cannot modify the sequence of route-maps in an existing map-list. To re-define the order of route-maps, you must create a new map-list.

To create a map list, use this procedure:

Task	Command
1. If you haven't already done so, define the route maps that you want to use to filter routes.	For more information about the **route-map** command and defining route maps, see "Creating Route Maps" on page 225.
2. Do one of the following: ■ If you are creating RIP route maps, enter router-rip mode. *or* If you are creating OSPF route maps, enter router-ospf mode.	■ **router rip** *or* ■ **router ospf**
3. Do one of the following: ■ If you are creating an import filter, enter import mode. *or* ■ If you creating an export filter, enter export mode.	■ **import** *or* ■ **export**
4. Add a route map to the map list then set the list to active or inactive. Repeat this command for all route maps that you want to add to the map-list.	**map-list** <route-map-list-number> **route-map** <route-map-number> **set** {**active** | **inactive**}

The following example creates a RIP import filter by adding route maps 1, 2, and 3 to the a map list number 20 and designates it as the active list to use for RIP imports.

```
cli:172.16.19.10:root# router rip
mode: router:rip
cli:172.16.19.10:router:rip# import
mode: router:rip:import
cli:172.16.19.10:router:rip:import# map-list 2 route-map 1
cli:172.16.19.10:router:rip:import# map-list 2 route-map 2
cli:172.16.19.10:router:rip:import# map-list 2 route-map  3 set active
cli:172.16.19.10:router:rip:import#
```

Route Filter Configuration Example

The following example creates two RIP import route-maps and adds them to map-list 1:

```
cli:172.16.19.10:root# router rip
mode: router:rip
cli:172.16.19.10:router:rip# import
mode: router:rip:import
cli:172.16.19.10:router:rip:import# route-map 10 permit
cli:172.16.19.10:router:rip:import:route-map(10)# match ip-address 172.16.19.0
255.255.255.0
cli:172.16.19.10:router:rip:import:route-map(10)# override metric 10
cli:172.16.19.10:router:rip:import:route-map(10)# route-map 11 deny
cli:172.16.19.10:router:rip:import:route-map(11)# match ip-address 172.16.19.23
 255.255.255.255
cli:172.16.19.10:router:rip:import:route-map(11)# map-list 1 route-map 11
cli:172.16.19.10:router:rip:import:route-map(11)# map-list 1 route-map 10
cli:172.16.19.10:router:rip:import:route-map(11)# map-list 1 set active
cli:172.16.19.10:router:rip:import:route-map(11)# show map-list 1

row count: 3

Template  Template  Row Status
          Order
--------  --------  --------------
       1         1  Active
      10         3  Active
      11         2  Active

cli:172.16.19.10:router:rip:import:route-map(11)#
```

12 CONFIGURING DHCP RELAY

This chapter provides information and procedures on how to configure DHCP relay on a cable interface and includes the following sections:

About DHCP Relay

DHCP is used within a DOCSIS- or EuroDOCSIS-compliant network to allocate IP addresses and to configure cable modems with other IP parameters.

DHCP Relay support on DOCSIS or EuroDOCSIS modules enables a cable interface (CMTS) to forward DHCP Requests from cable modems, CPE devices, MTA devices, and other IP hosts to a DHCP server. The DHCP server may reside:

- Externally, on a system other than the Cuda 12000 that has the cable interface you are configuring.
- Internally, on the same Cuda 12000 that has the cable interface that you are configuring.

Note: You may configure the CMTS to forward DHCP Requests to up to 32 servers using DHCP Policies. For more information, see "Configuring DHCP and BOOTP Policies" on page 253.

Gateway addresses are used by the DHCP Relay to request a specific subnet for the host and cable modem.

Configuration is accomplished within interface cable <c/s/i> mode using the **dhcp-relay** command. Using this command, you can configure the following functions:

- Enable or disable DHCP Relay
- Configure the following gateway addresses:
  - **CPE/IP Host Gateway Address** — The Host Gateway address that the DHCP Relay requests on behalf of the host. This is the same address as the Gateway Address configured on the interface. When a DHCP request is received and it is from the host, then the Host Gateway Address is used by the DHCP Relay.
  - **Cable Modem (CM) Gateway Address** — The CM Gateway address that the DHCP Relay requests on behalf of the cable modem. This is the same address as the Gateway Address configured on the interface. When a DHCP Request is received and it is from the cable modem, then the CM Gateway Address is used by the DHCP Relay.

- **MTA Gateway Address** — The MTA Gateway address that the DHCP Relay requests on behalf of the MTA device. This is the same address as the Gateway Address configured on the interface.When a DHCP Request is received and it is from the MTA device, then the MTA Gateway Address is used by the DHCP Relay.

- Enable or disable agent options.

Displaying DHCP Relay Configuration

You can display the current DHCP relay configuration using the `show` **dhcp-relay** command from within interface cable <c/s/i> mode, as shown in the following example:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-relay
dhcp-relay                              enable
Add Agent Options                       enable
Drop Mismatch                          disable
Max. Pkt. Len.                             576
Relay Mode                             replace



Server Address
----------------

giAddresses:
CM                                  201.1.1.1
CPE                                 201.1.2.1
MTA                                   0.0.0.0

cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Table 12-1 describes the fields shown in the display.

Table 12-1 DHCP-Relay Display Fields

Field	Description
dhcp-relay: [enabled *or* disabled]	Indicates whether or not the DHCP Relay is enabled on this cable interface.
Add Agent Options	Indicates whether Agent Options are enabled on this cable interface.
Drop Mismatch	Indicates whether the Drop Mismatch options is enable or disabled.
Max Pkt Len	Maximum packet length allowed to be relayed.
Relay Mode	DHCP Relay mode configured on this interface.
giAddresses	Indicates the IP gateway that is used by: ■ Cable modems (CM) ■ CPE devices attached to this cable interface ■ MTA devices

To configure which DHCP servers the CMTS forwards requests to, use the **dhcp-policy** command. Using DHCP Policies, you can configure the DHCP Relay agent to forward requests to a list of up to 32 DHCP servers. For more information, see *"Configuring DHCP and BOOTP Policies"* on page 253.

Configuring DHCP Relay Options

The Cuda 12000 allows you to enable and configure DHCP Relay functionality on each IP interface so that the interface can forward DHCP requests to a central DHCP server. You must enable DHCP Relay on a select CMTS interface to dynamically allocate network addresses to the attached cable modems. You must also enable DHCP agent options for that interface if you plan to place cable modems, attached CPE hosts and MTA devices on different networks.

Use the following procedure to configure DHCP Relay on a cable interface.

Task	Command
1. Enter configuration mode for the the interface on which you want to DHCP Relay.	**interface** <c/s/i>
2. Enable DHCP Relay on the current interface. Disabling DHCP Relay prevents the DHCP server from assigning IP addresses to hosts on the interface.	**dhcp-relay** {**enable | disable**}
3. Show the available IP interfaces on the selected physical interface. You must choose one of these interfaces as a gateway for cable modems and CPE hosts *(next)*. If you have not yet done so, you must add the IP address of the gateway *(that you want the devices on this interface to use)* to the interface using the **ip address** command.	**show ip-address**
4. If you plan to specify different gateways for cable modem, CPE devices and MTA devices to use, you must enable DHCP agent options.	**dhcp-relay add-agent-options enable** *Note that this option must be enabled when using the FastFlow Broadband Provisioning Manager.*
5. Configure the gateway that you want the CPE/IP hosts on this interface to use. Select one of the IP interfaces defined on the current physical interface.	**dhcp-relay cpe-gateway** <gateway address>

Task	Command
6. If you are configuring a cable interface, then configure the gateway that you want cable modems on this interface to use.	**dhcp-relay cm-gateway** <gateway address>
7. If you are configuring an MTA interface, then configure the gateway that you want MTA devices on this interface to use.	**dhcp-relay mta-gateway** <gateway address>
8. Verify the current dhcp-relay parameters for the current interface.	**show dhcp-relay**

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-relay enable
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-relay add-agent-options enable
cli:192.168.208.3:interface:cable:csi(1/1/1)# show ip address
Chassis/Slot/Interface      1/1/1

row count: 2

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
201.1.1.1        255.255.255.0       8781825      Other
201.1.2.1        255.255.255.0       8781825    Primary

cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-relay cm-gateway 201.1.2.1
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-relay cpe-gateway 201.1.2.1
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-relay mta-gateway 201.1.2.1
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-relay
dhcp-relay                           enable
Add Agent Options                    enable
Drop Mismatch                       disable
Max. Pkt. Len.                          576
Relay Mode                          replace

Server Address
----------------
giAddresses:
CM                                201.1.2.1
CPE                               201.1.2.1
MTA                               201.1.2.1
```

Specifying DHCP Servers

You must specify the DHCP server to which you want the cable interface to forward DHCP Requests. The DHCP server is configured on a per interface basis. You may add up to 32 DHCP servers.

If a DHCP server is not configured, then the DHCP server drops all DHCP requests as it does not know where to forward them.

DHCP servers fall into two categories:

- **External** — DHCP servers that reside on systems other than the Cuda 12000 that has the cable interface that you are configuring. DHCP messages are forwarded over the network to a remote, external DHCP server.
- **Internal** — A FastFlow Broadband Provisioning Manager DHCP server that resides on the same Cuda 12000 that has the cable interface you are configuring (that is, the local Cuda 12000). DHCP requests are forwarded internally to the DHCP server.

Specifying External DHCP Servers

To specify an external DHCP server, you configure the DHCP Relay Agent on the cable interface to point to the IP address of a remote DHCP server. To do so, perform the following tasks.

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Specify an external DHCP server.	**dhcp-policy default permit** <ip-address>

Example

The following example configures cable interface 1/1/1 to forward DHCP messages to the DHCP server at address 201.1.13.1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy
default permit 201.1.13.1
```

Specifying the Internal DHCP Server

To specify the internal FastFlow BPM DHCP server, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Specify the internal DHCP server.	**dhcp-policy default permit forward-internal**

Example

The following example configures cable interface 1/1/1 to forward DHCP messages to the internal FastFlow BPM DHCP server:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy
default permit forward-internal
```

DHCP and BOOTP Policies

You can use Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) policies to control which devices obtain IP addresses and which DHCP and BOOTP servers allocate those addresses. This section provides information and procedures about configuring DHCP and BOOTP policies on the Cuda 12000 and includes the following sections:

- About DHCP Policies
- About BOOTP Policies
- Configuring DHCP and BOOTP Policies
- Configuring Default Policies

About DHCP Policies

A DOCSIS- or EuroDOCSIS-compliant network uses DHCP for dynamic assignment of IP addresses. A DHCP server allocates addresses and other IP operational parameters to requesting cable modems, CPE devices and MTA devices. DOCSIS and EuroDOCSIS modules serve as cable modem termination systems and, as such, also function as DHCP relay agents. As relay agents, these cable interfaces relay DHCP requests and responses between the DHCP server and cable modems, CPE devices and MTA devices.

DHCP policies allow you to control and restrict the forwarding of DHCP requests. Specifically, DHCP policies allow matching on several parameters in the DHCP packet. It then uses the result of this matching to determine which list of servers to forward the packet to; or it can reject (drop) the packet to deny the requesting client an address.

DHCP policies allow you to do the following:

- Prevent select modems, CPE devices and MTA devices from obtaining IP addresses.
- Direct DHCP requests to particular DHCP servers based on whether the request originated from a modem, CPE device or MTA device.
- Direct DHCP requests to particular DHCP servers based on the cable modem's, CPE's or MTA's MAC address.
- Direct DHCP requests to particular servers based on which interface it was received.

For example, you can configure the system to match on the DHCP packet to determine whether the request originated from a cable modem, a CPE, a MTA device, a specific interface, or a specific MAC address; wildcards can be used to match portions of a MAC address. In the event of a match, you can configure the DHCP relay agent to forward the request to a list of up to 32 DHCP servers, or configure the agent to drop the request.

If there are no policies defined, or a DHCP packet does not match any existing policy, the default policy is used to determine if the packet is dropped or forwarded to a list of up to 32 DHCP servers. The Cuda 12000 ships with a default policy to deny *(drop)* DHCP requests that do not match any other policy. Note that while other DHCP policies are interface-specific, the default DHCP policy is module-wide—it provides default behavior for all interfaces on the module. This default policy can be modified but not deleted.

About BOOTP Policies

BOOTP is a protocol that allows diskless workstations to boot off of a network server, called a BOOTP server. You can configure the cable interface to deny (drop) a matching BOOTP request or permit it to be forwarded to a list of BOOTP servers.

Configuring DHCP and BOOTP Policies

DHCP Policies determine the DHCP servers to which a CMTS interface forwards DHCP requests from attached cable modems, CPE devices and MTA devices.

BOOTP Policies determine the BOOTP servers to which a CMTS interface forwards BOOTP requests from attached cable modems and diskless workstations.

You configure DHCP policies using the **dhcp-policy** command. You configure the BOOTP policies using the **bootp-policy** command.

Defining Policies for a Cable Interface

Using the **dhcp-policy** or **bootp-policy** command, you can configure the cable interface to forward DHCP or BOOTP requests.

The following table describes the parameters that you set to configure DHCP policies:

Table 12-2 DHCP Policy Parameters

Parameter	Description
Index Number	Determines the sequence in which a DHCP request is compared to each policy. You assign this number when defining the policy. The request is applied to the policy with the lowest index first, then precedes incrementally. Upon finding a match, the action defined for the policy is taken, and no further policies are applied.
Policy Server List	A list of IP addresses to which you want the current cable interface to forward DHCP packets.

Parameter	Description
Match Criteria	DHCP Policies allow matching on several parameters in the DHCP packet, including: ■ Agent-Options – Determines whether the DHCP request is from a cable modem, CPE device, or MTA device. ■ Cable Modem MAC Address – Allows you to match on the cable modem MAC address contained in the request. ■ Interface – Enables you to match on the specific interface on which the DHCP offer was received. ■ MAC Address – Allows you to match on the source MAC address of the cable modem. You can also wildcard any or all octets of the MAC address.
Policy Action	Specifies the action that you want the system to take upon finding a matching DHCP request. You can configure the interface to do the following: ■ Permit the packet to be forwarded to up to 32 DHCP servers ■ Deny (drop) the packet without forwarding it
Forward-Internal	Specifies that the current cable interface forwards DHCP requests internally (meaning, to a DHCP server on the local Cuda 12000). Optionally, you can specify the **disable** keyword to disable internal forwarding.

The following table describes the parameters that you set to configure BOOTP policies:

Table 12-3 BOOTP Policy Parameters

Parameter	Description
Index Number	Determines the sequence in which a BOOTP request is compared to each policy. You assign this number when defining the policy. The request is applied to the policy with the lowest index first, then precedes incrementally. Upon finding a match, the action defined for the policy is taken, and no further policies are applied.
Policy Server List	A list of IP addresses to which you want the current cable interface to forward BOOTP messages.
Match Criteria	Matching MAC addresses of BOOTP clients. These devices are either permitted to send BOOTP messages to specified BOOTP servers or denied permission to send BOOTP messages. Using masks, you can specify a range of MAC addresses.
Policy Action	The action that you want the system to take upon finding a matching BOOTP request. You can configure the interface to do the following: ■ Permit the BOOTP message to be forwarded to BOOTP servers. ■ Deny (drop) the BOOTP message from being forwarded to BOOTP servers.

To configure a DHCP Policy and a BOOTP policy, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Create a DHCP Policy.	**dhcp-policy** {<policy-index> | **default**} {**deny** | **permit**} {<ip-address>... | **forward-internal** [**disable**] } [**agent-option** {**cm** | **cpe**} | **cmmac** <mac-address> | **interface** <c/s/i> | **mac** <mac-address> [**mask** <mask>]] [**vendor-class-id** {cm | mta}] [**description** <string>]
3. Create a BOOTP policy.	**bootp-policy** {<policy-index> | **default**} {**deny mac** <mac-address> [**mask** <mask>] ... | **permit** <ip-address>... **mac** <mac-address> [**mask** <mask>] ...} [**description** <string>]

To display the DHCP or BOOTP policies currently configured on a cable interface, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Display all DHCP policies.	**show dhcp-policy**
3. Display all BOOTP policies.	**show bootp-policy**
4. Display a specified DHCP policy.	**show dhcp-policy** <policy-index>
5. Display a specified BOOTP policy.	**show bootp-policy** <policy-index>
6. Display the default DHCP policy.	**show dhcp-policy default**
7. Display the default BOOTP policy.	**show bootp-policy default**

To remove DHCP or BOOTP policies from the current cable interface, perform the following tasks within interface cable <c/s/i> mode:

Task	Command
1. Remove all DHCP policies from the current cable interface.	**no dhcp-policy all**
2. Remove all BOOTP policies from the current cable interface.	**no bootp-policy all**
3. Remove a specified DHCP policy from the current cable interface.	**no dhcp-policy** <policy number>
4. Remove a specified BOOTP policy from the current cable interface.	**no bootp-policy** <policy number>

Configuring Default Policies

To modify the default DHCP or BOOTP policy used on all interfaces of a specific DOCSIS module, enter interface cable <c/s/i> mode for one of the interfaces on the selected DOCSIS module and perform the following task:

Task	Command
1. Modify the default DHCP policy. *Note that you do not define matching criteria for the default policy. You simply configure it to permit forwarding to up to 32 DHCP servers, or drop the packet.*	**dhcp-policy default** {**permit** <ip-address>... | **deny**}
2. Modify the default BOOTP policy. *Note that you do not define matching criteria for the default policy. You simply configure it to permit forwarding to BOOTP servers, or drop the message.*	**bootp-policy default** {**permit** <ip-address>... | **deny**}

To display the currently configured default DHCP or BOOTP policy, enter interface cable <c/s/i> mode for one of the interfaces on the selected DOCSIS module and perform the following tasks:

Task	Command
1. Display the default DHCP policy.	**show dhcp-policy default**
2. Display the default BOOTP policy.	**show bootp-policy default**

Note that you cannot delete the default DHCP or BOOTP policy.

DHCP Policy Configuration Examples

This section contains examples illustrating how to configure DHCP policies for a specified cable interface.

The following example configures the DHCP relay agent to forward DHCP requests internally to the local FastFlow BPM DHCP server:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy
default permit forward-internal
```

The following example configures the DHCP relay agent on cable interface 1/1/1 to deny and drop any DHCP request from cable modem 03:02:09:a4:90:12:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy 1 deny
cmmac 03:02:09:a4:90:12
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-policy 1

Index                                       1
Mac Address
Mac Mask
Cable Modem Mac              03:02:09:a4:90:12
Policy Action                            deny
Policy Server List
Description
CM                                      False
CPE                                     False
C/S/I/P                          0 / 0 / 0 / 0

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The following example configures the cable interface to forward all DHCP requests arriving on interface 1/1/1 to servers 102.12.1.12 and 172.16.19.3:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy 2
permit 102.12.1.12 172.16.19.3 interface 1/1/1
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-policy 2
Index                                       2
Mac Address
Mac Mask
Cable Modem Mac
Policy Action                            permit
Policy Server List         102.12.1.12 172.16.19.3
Description
CM                                        False
CPE                                       False
C/S/I/P                          1 / 1 / 1 / 2

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The following example configures cable interface 1/1/1 to forward all DHCP requests from CPE devices to server 101.1.13.5:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy 3
permit 101.1.13.5 agent-option cpe
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-policy 3
Index                                       3
Mac Address
Mac Mask
Cable Modem Mac
Policy Action                            permit
Policy Server List         101.1.13.5
Description
CM                                        False
CPE                                        True
C/S/I/P                          0 / 0 / 0 / 0

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The following example configures the cable interface to forward all DHCP requests containing a source MAC address of 09:08:a4:95:2e:3a to server 101.1.1.1:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-policy 4
permit 101.1.1.1 mac 09:08:a4:95:2e:3a mask 00:00:00:00:00:00
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-policy 4
Index                                    4
Mac Address                09:08:a4:95:2e:3a
Mac Mask                   00:00:00:00:00:00
Cable Modem Mac
Policy Action                        permit
Policy Server List         101.1.1.1
Description
CM                                    False
CPE                                   False
C/S/I/P                        0 / 0 / 0 / 0

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

When matching on the source MAC address of the request *(using the mac keyword parameter)*, you can wildcard any or all octets of the MAC address by masking each wildcard octet with "FF." For example, the following MAC address/mask pair would match all addresses starting with 00:02:09:

MAC address: 00:02:09:00:00:00
Mask: FF:FF:FF:00:00:00

13 CONFIGURING DHCP AUTHORITY

This chapter provides instructions on how to configure DHCP authority and includes the following sections:

About DHCP Authority

DHCP authority is a security feature that prevents spoofing *(unauthorized use)* of DHCP assigned IP addresses. Spoofing occurs when a host uses an IP address that was dynamically assigned to another host via the Dynamic Host Configuration Protocol (DHCP). DHCP authority prevents spoofing of IP addresses by ensuring that IP addresses are only used by the specific cable modems and CPEs to which they are assigned.

Configured on an interface basis, DHCP authority ensures that dynamically assigned IP addresses are used by their original host by tagging Address Resolution Protocol (ARP) entries within the ARP cache for a specified interface.

This DHCP Authority ARP entry tagging process operates as follows:

- Upon booting, the client (such as a cable modem or CPE device) requests an IP address from the DHCP server. The DHCP relay agent operating on the interface to which the client is attached, forwards the request to the DHCP server.
- Based on the subnet configuration within the provisioning server, the DHCP server responds with a DHCP offer containing the IP address that the client should use.
- After receiving the IP address, the client sends a DHCP request back to the DHCP server.
- The DHCP server then sends a DHCP acknowledgement to the client through the DHCP relay.

- When the DHCP relay agent sees this acknowledgement, it then checks to verify whether the IP address falls within a DHCP authority range configured on the interface, and one of the following actions occur:

  If the address does fall within a preconfigured DHCP authority range and DHCP Authority is enabled for that interface, an ARP entry is added to the ARP cache for that interface and tagged as being assigned through DHCP. This tag is shown as type "Other" when viewing the ARP cache for that interface and ensures that specific IP address only maps to that specific MAC address.

  or

  If there is no DHCP Authority range, the entry is simply added to the ARP cache and labelled as type "Dynamic" when the ARP mapping is learned.

This feature is termed DHCP authority because those tagged as being assigned through DHCP take precedence over dynamically assigned *(*non-DHCP tagged*)* ARP entries. In the ARP cache entries that follow, those labelled as *other* are protected by DHCP authority, those labelled as dynamic are not:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show arp

row count: 4
IP Address       MAC Address        Type
---------------- ------------------ ----------
192.168.19.51    00:10:95:04:0a:c4     dynamic
192.168.19.52    00:10:95:04:0a:b7     dynamic
192.168.19.55    00:10:95:01:ef:d8       other
192.168.19.56    00:a0:73:69:39:65       other

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Enabling DHCP Authority

To enable or disable DHCP authority on an interface, perform the following tasks:

Task	Command
1. Enter interface mode.	**interface** <c/s/i>
2. Enable DHCP Authority on the current interface.	**dhcp-authority** {**enable** | **disable**}

Example

The following example enables DHCP authority on cable interface 1/1/1, then uses the **show dhcp-authority** command to verify the configuration:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-authority enable
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-authority

Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------

DHCP Authority Status                  enable

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Configuring DHCP Authority Ranges

The DHCP Authority ranges that you define for an interface dictate which addresses are protected by the authority feature. These DHCP authority IP address ranges that you define must fall within the range of IP addresses as allowed by the IP interface *(*as dictated by the network mask for that IP interface*)*.

For example, if the physical interface has an IP interface of 172.16.19.1 255.255.255.0 installed, you can define a DHCP Authority range from 172.16.19.2 to 172.16.19.254, or any subset of that IP address range. Note that you can define up to 200 IP address ranges per physical interface.

You define a DHCP authority range by performing the following tasks:

Task	Command
1. Enter interface type mode.	**interface** <c/s/i>
2. Configure a DHCP authority range on the current interface.	**dhcp-authority** <index number> **start** <ip address> **end** <ip address>

The DHCP authority ranges take effect upon the next DHCP server exchange with the client. This means that after you configure a range, you should reboot the client so that the ARP entry for that client is updated.

Example

The following example configures a DHCP authority range on cable interface 1/1/1 that protects all addresses that fall within 172.16.19.10 to 172.16.19.20, the uses the **show dhcp-authority** command to verify the configuration:

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-authority 1 start
172.16.19.10 end 172.16.19.20
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority
row count: 1
Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------
               1 172.16.19.10     172.16.19.20          1

DHCP Authority Status                enable
```

Removing DHCP Authority Ranges

You can remove DHCP Authority ranges using the **no dhcp-authority** command. You may want to do so if you no longer require the range, or if you want to redefine the range. Note that you cannot modify DHCP Authority ranges, so if you want to redefine a range, you must delete it and then recreate it with the new configuration.

To remove a DHCP authority range from an interface, perform the following task within interface configuration mode:

Task	Command
1. Enter interface configuration mode.	**interface** <c/s/i>
2. Remove a DHCP authority range from the current interface.	**no dhcp-authority** <index number>

Example

The following example removes DHCP authority range 1 from interface 1/1/1:

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority

row count: 1

Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------
               1 172.16.19.10     172.16.19.20          1

DHCP Authority Status                enable

cli:192.168.208.3:interface:cable:csi(1/1/1)# no dhcp-authority 1
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-authority


Range Number     Lower Range      Upper Range      Status
---------------- ---------------- ---------------- ------

DHCP Authority Status                enable
```

DHCP Authority Configuration Examples

In the following example, cable interface 1/1/1 has an IP interface of 192.168.19.50:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip address
Chassis/Slot/Interface      1/1/1

row count: 1

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
192.168.19.50    255.255.255.0       8781825      Other
```

In the following example, this IP address is configured as the cable modem gateway. The ARP cache for that interface shows that all DHCP addresses were provided dynamically and are not protected by DHCP authority:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show arp

row count: 12

IP Address       MAC Address        Type
---------------- ------------------ ----------
192.168.19.51    00:10:95:04:0a:c4     dynamic
192.168.19.52    00:10:95:04:0a:b7     dynamic
192.168.19.53    00:90:96:00:29:6d     dynamic
192.168.19.54    00:10:95:04:0a:c3     dynamic
192.168.19.55    00:10:95:01:ef:d8     dynamic
192.168.19.56    00:a0:73:69:39:65     dynamic
192.168.19.57    00:90:83:36:82:f1     dynamic
192.168.19.58    00:90:96:00:39:f9     dynamic
192.168.19.59    00:90:96:00:39:7f     dynamic
192.168.19.60    00:10:95:01:f0:05     dynamic
192.168.19.61    00:90:83:32:9f:8c     dynamic
192.168.19.62    00:90:83:36:82:ee     dynamic

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The following example protects all IP addresses within the address range of 192.168.19.51 to 192.168.19.55 using the **dhcp-authority** command.

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# dhcp-authority 1 start 192.168.19.
51 end 192.168.19.55
cli:172.16.19.10:interface:cable:csi(1/1/1)# show dhcp-authority

row count: 1

Range Number      Lower Range       Upper Range       Status
---------------- ---------------- ---------------- ------
               1 192.168.19.51    192.168.19.55          1

DHCP Authority Status                  enable

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The next time that the attached cable modems reinitialize, those cable modems assigned IP addresses that fall within the DHCP Authority range are tagged by the DHCP server as type *Other*. Those entries shown as type *Other* are protected by the DHCP Authority feature. Entries shown as "dynamic" are fixed IP addresses.

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show arp

row count: 12

IP Address       MAC Address        Type
---------------- ------------------ ----------
192.168.19.51    00:10:95:04:0a:c4       other
192.168.19.52    00:10:95:04:0a:b7       other
192.168.19.53    00:90:96:00:29:6d       other
192.168.19.54    00:10:95:04:0a:c3       other
192.168.19.55    00:10:95:01:ef:d8       other
192.168.19.56    00:a0:73:69:39:65     dynamic
192.168.19.57    00:90:83:36:82:f1     dynamic
192.168.19.58    00:90:96:00:39:f9     dynamic
192.168.19.59    00:90:96:00:39:7f     dynamic
192.168.19.60    00:10:95:01:f0:05     dynamic
192.168.19.61    00:90:83:32:9f:8c     dynamic
192.168.19.62    00:90:83:36:82:ee     dynamic

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

14 CONFIGURING IP

This chapter provides information on how to configure Internet Protocol (IP) routing protocols on the Cuda 12000 and describes the following functions:

Configuring IP Addresses

Configuring IP addresses involves setting values for the following parameters:

- **IP Address —** Enter the IP address that you want to assign to the selected physical interface (chassis/slot/interface) or the loopback interface, which is a logical IP interface.
- **Network Mask Address** — Enter the network mask for that network.

You add an IP address to an interface using the **ip address** command within interface configuration mode. If you specify an IP address that does not exist, the **ip address** command creates the address, then enters configuration mode for the IP interface; if the IP address already exists, it simply enters configuration mode for the address. In either case, the IP address is then displayed within the command prompt.

Within IP address configuration mode, you can configure RIP interface parameters using the **ip rip** command and OSPF interface parameters using the **ip ospf** command. You cannot configure RIP and OSPF interface parameters for loopback interfaces.

The loopback interface that you can configure through the CLI differs from the standard IP loopback interface, which has a fixed address of 127.0.0.1. One of the uses of the configurable loopback interface is for opening in-band connections (such as telnet sessions) to the Cuda 12000. Users can specify the IP address of the loopback interface to open connections, thereby eliminating the need to specify the IP address of a physical interface (such as a CMTS interface). You can also configure static routes via the loopback interface to act as backup routes to those configured via physical interfaces. Note that the <c/s/i> of the interface appears as 131/1/1 in **show** command output displays.

You add an IP address to a selected physical interface or the loopback interface, and then enter configuration mode for that IP interface by performing the following tasks:

Task	Command
1. Enter configuration mode for the selected interface	**interface** {<c/s/i> | **loopback**}
2. Add the IP address.	**ip address** <ip-address> <mask>

Example 1

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip address 204.142.19.2 255.255.255.0
cli:172.16.19.10:interface:cable:csi(1/1/1):ip-address(204.142.19.2)# show ip
address
Chassis/Slot/Interface      1/1/1

row count: 2

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
192.168.19.50    255.255.255.0       8781825      Other
204.142.19.2     255.255.255.0       8781825    Primary

cli:172.16.19.10:interface:cable:csi(1/1/1):ip-address(204.142.19.2)#
```

Example 2

```
cli:192.168.208.3:root# interface loopback
mode: interface:loopback
cli:192.168.208.3:interface:loopback# show ip address
Chassis/Slot/Interface      131 / 1 / 1
row count: 0
IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------

cli:192.168.208.3:interface:loopback# ip address 205.1.1.1 255.255.255.0
cli:192.168.208.3:interface:loopback# show ip address
Chassis/Slot/Interface      131 / 1 / 1
row count: 1
IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
205.1.1.1        255.255.255.0    1099169793    Primary
```

Viewing IP Interfaces

You can use the following commands to view IP interfaces:

- **show ip address**: Use this command within interface configuration mode to view a list of IP addresses added to that current physical interface.
- **show ip interface**: Shows IP address information and additional information about the physical interface on which the IP interface resides:
  - When issued within interface configuration mode, displays only information relevant to the current interface.
  - When issued within slot mode, displays all IP interfaces within the current slot.
  - When issued in any other mode, displays all IP interface configured throughout the system.

Example 1

The following example displays all IP addresses added to cable interface 1/1/1:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show ip address
Chassis/Slot/Interface      1/1/1

row count: 2

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
201.1.1.1        255.255.255.0       8781825      Other
201.1.2.1        255.255.255.0       8781825    Primary
```

Example 2

The following example displays IP interface information for interface 1/1/1:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show ip interface
Chassis/Slot/Interface     1 / 1 / 1 (8781825)
Description                CATV MAC: Broadcom BCM3210
Admin Status                            up
Oper Status                             up
Mtu                                   1500  (bytes)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
IP Address                         201.1.1.1
Net Mask                       255.255.255.0
Interface                            8781825
Priority                               Other
RIP configuration:
Send Version                           RIPv1
Receive Version                        RIPv1
Cost                                       1
Authentication ON                      False
Authentication Type        No Authentication
Authentication Key Id                      0
Send Default Only                      False
Send Default Also                      False
Default Cost                               0
Accept Default                          True
Accept Host Route                       True
Split Horizon                           True
Poisoned Reverse                        True
Status                                Active
OSPF configuration:
- - - - - - - - - - - - - - - - - - -- - - - - - - - - - - - - - - -
IP Address                         201.1.2.1
Net Mask                       255.255.255.0
Interface                            8781825
Priority                             Primary
RIP configuration:

OSPF configuration:
```

Deleting IP Addresses

You may want to remove an IP address from a physical interface or the loopback interface when you no longer need the associated network or you want to assign the address to a new interface.

You remove an IP network from a physical interface or loopback interface by performing the following tasks.

Task	Command
1. Enter configuration mode for the selected interface	**interface** {<c/s/i> | **loopback**}
2. Delete the IP address.	**no ip address** <ip-address>
3. ***Optional***. Verify that you have removed the IP interface.	**show ip interface**

Example

```
cli:192.168.208.3:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show ip address
Chassis/Slot/Interface      1/11/1

row count: 1

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------
205.2.3.1        255.255.255.0      11337729    Primary

cli:192.168.208.3:interface:ethernet:csi(1/11/1)# no ip address 205.2.3.1
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show ip address
Chassis/Slot/Interface      1/11/1

row count: 0

IP Address       Net Mask         Interface  Priority
---------------- ---------------- ---------- ----------

cli:192.168.208.3:interface:ethernet:csi(1/11/1)#
```

Displaying the Routing Table

You can display the contents of the routing table using the **show ip** command within any mode, as shown in the following example:

```
cli:192.168.208.3:root# show ip

row count: 8

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Net Mgmt Remote 133.1.1.0       255.255.255.0   201.1.1.10           1 1/1/1
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1          0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1            0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.2            1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2            0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1
```

Note that:

- The loopback interface is identified with the c/s/i of 131/1/1.
- Static routes have a protocol type of "Net Mgmt."
- Routes to directly connected networks via local interfaces have a protocol type of "Local."

You can narrow the output to include only what you need by entering the **include** command such as in the following: **show ip | include "Net Mgmt"** will narrow the output to only static routes. Note that the string you specify is case-sensitive. Strings containing spaces must be enclosed in quotes.

The following example shows the output of the **show ip** command to include only static routes contained in the table:

```
cli:192.168.208.3:root# show ip | include "Net Mgmt"
Net Mgmt Remote 133.1.1.0       255.255.255.0   201.1.1.10           1 1/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.2            1 131/1/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1
```

Configuring Static Routes

You can manually add static routes to the Cuda 12000 routing table. Static routes take precedence over dynamically-learned routes to the same destination. These routes are useful in network environments where no routing protocol is used, or to override select routes discovered using a routing protocol.

Configuring static routes consists of the following tasks:

- Adding Static Routes
- Deleting Static Routes
- Adding the Default Route
- Deleting the Default Route

Adding Static Routes

To add a static route, configure the following parameters:

Table 14-1 Static Route Parameters

Parameter	Description
Destination	IP address of the destination network, subnetwork or host.
Network Mask	Network Mask assigned to the destination IP address.
Gateway	IP address of the next hop (gateway) used to reach the destination.
Metric	Assigns a metric to this static route. The metric is a number that is used to select the route when multiple routes to the same destination exist. The route with the lowest metric is selected. The default is 1.

Perform the following tasks to add a static route to the routing table:

Task	Command
1. Enter configuration mode for the interface on which you want to add the route.	**interface** {<c/s/i> | **loopback**}
2. Add the static route.	**ip route** <dest-network> <mask> <gateway-ip-address> [<metric>]

Example

```
cli:192.168.208.3:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# ip route 222.2.1.0
255.255.255.0 222.2.2.1
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show ip

row count: 8

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1          0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1            0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.3            1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Net Mgmt Remote 222.2.1.0       255.255.255.0   222.2.2.1            1 1/11/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2            0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1
```

Deleting Static Routes

Static routes remain in the routing table until you remove them. You may want to remove the route if it is no longer needed, or if you prefer that the system discover the route dynamically.

You delete a static route from the routing table by performing the following task:

Task	Command
1. Enter configuration mode for the interface on which you want to delete the route.	**interface** {<*c/s/i*> | **loopback**}
2. Display the routing table.	**show ip**
3. Delete the static route.	**no ip route** <dest-network> <mask> [<gateway-ip-address>] *Note that you specify the IP address of the gateway only if a route with a duplicate destination network and mask exists.*

Example

```
cli:192.168.208.3:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show ip

row count: 8

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ -------------- -------------- --------------- ------ ------
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1          0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1            0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.3            1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Net Mgmt Remote 222.2.1.0       255.255.255.0   222.2.2.1            1 1/11/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2            0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# no ip route 222.2.1.0
255.255.255.0
cli:192.168.208.3:interface:ethernet:csi(1/11/1)# show ip

row count: 7

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ -------------- -------------- --------------- ------ ------
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1          0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1            0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.3            1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2            0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1

cli:192.168.208.3:interface:ethernet:csi(1/11/1)#
```

Adding the Default Route

The default route is a special kind of static route. When the Cuda 12000 must forward a packet, but it cannot determine the route to the packet's destination, the Cuda 12000 forwards the packet to the next hop associated with the default route.

Unlike other static routes, you only have to specify the IP address of the next hop (gateway) when adding the default route.

To add the default route, perform the following task:

Task	Command
From any mode, add the default route.	**ip route default** <gateway-ip-address>

Example

```
cli:192.168.208.3:root# ip route default 201.1.1.10
cli:192.168.208.3:root# show ip

row count: 8

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ -------------- -------------- -------------- ------ ------
Net Mgmt Remote 0.0.0.0         0.0.0.0         201.1.1.10          1 1/1/1
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1         0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1           0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.3           1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1           0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1           0 1/1/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2           0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1           1 1/11/1
```

When you display the routing table, note that the default route has a Destination of 0.0.0.0 and a Net Mask of 0.0.0.0.

Deleting the Default Route

To add the default route, perform the following task:

Task	Command
From any mode, delete the default route.	**no ip route default**

Example

```
cli:192.168.208.3:root# show ip

row count: 8

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Net Mgmt Remote 0.0.0.0         0.0.0.0         201.1.1.10           1 1/1/1
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1          0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1            0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.3            1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2            0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1

cli:192.168.208.3:root# no ip route default
cli:192.168.208.3:root# show ip

row count: 7

Protocol Route  Destination     Net Mask        Next Hop        Metric C/S/I
         Type
-------- ------ --------------- --------------- --------------- ------ ------
Local    Local  155.144.1.0     255.255.255.0   155.144.1.1          0 1/8/1
Local    Local  199.3.1.0       255.255.255.0   199.3.1.1            0 131/1/1
Net Mgmt Remote 199.3.2.0       255.255.255.0   199.3.1.3            1 131/1/1
Local    Local  201.1.1.0       255.255.255.0   201.1.1.1            0 1/1/1
Local    Local  201.1.2.0       255.255.255.0   201.1.2.1            0 1/1/1
Local    Local  222.2.2.0       255.255.255.0   222.2.2.2            0 1/11/1
Net Mgmt Remote 222.2.3.0       255.255.255.0   222.2.2.1            1 1/11/1
```

Managing the Address Resolution Protocol (ARP)

On a physical network on which devices have Media Access Control (MAC) addresses (for example, Ethernet), ARP is used to map the MAC addresses to IP addresses. Managing ARP consists of the following tasks:

- Displaying the ARP cache
- Adding ARP entries
- Deleting ARP entries
- Configuring the ARP Timeout
- Clearing the ARP Cache

Each host on an IP network has two addresses:

- **MAC address** — Identifies the host at layer 2, the datalink layer, of the OSI model.
- **IP address** — Identifies the host at layer 3 of the OSI model and indicates the network to which it belongs.

To forward packets to a host on an IP network, an interface must know the IP address of the target host. Once the interface learns IP-to-MAC address mapping for a host, it stores this information in its ARP cache.

Displaying the ARP Cache

Use the following procedure to display the ARP cache for a selected interface

Task	Command
1. Enter configuration mode for the selected interface.	**interface** <c/s/i>
2. Display the ARP cache for the selected interface.	**show arp**

Example

The following example displays the contents of the ARP cache maintained by cable interface 1/1/1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show arp

row count: 10

IP Address       MAC Address        Type
---------------- ------------------ ----------
192.168.19.51    00:10:95:04:0a:c3     dynamic
192.168.19.52    00:90:96:00:39:7f     dynamic
192.168.19.53    00:10:95:01:ef:d8     dynamic
192.168.19.54    00:a0:73:69:39:65     dynamic
192.168.19.55    00:10:95:04:0a:b7     dynamic
192.168.19.56    00:90:96:00:29:71     dynamic
192.168.19.57    00:90:96:00:29:6d     dynamic
192.168.19.58    00:10:95:01:f0:05     dynamic
192.168.19.59    00:90:96:00:39:f9     dynamic
192.168.19.60    00:90:83:32:9f:8c     dynamic

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

For each ARP entry, the **show arp** command output displays the IP address of a device, the MAC address of that device, and the type of ARP entry (dynamic, static, or other). Dynamic entries are created automatically by the ARP protocol, while static entries are created using the **add arp** command. Static entries can be created on Ethernet interfaces only. ARP entries that have a type of "other" are protected by the DHCP authority feature.

Adding ARP Entries

System interfaces can learn host addresses by sending out ARP requests. Optionally, you can manually define these address mappings for a selected interface by adding a static IP-to-MAC address entry to the ARP cache of a selected interface.

Adding an ARP entry involves specifying both the IP address and the MAC address that you want to map to each other.

You can add static ARP entries to Ethernet interfaces only.

You add an ARP entry to a selected interface by performing the following tasks:

Task	Command
1. Enter configuration mode for the selected Ethernet interface.	**interface** <c/s/i>
2. Add the IP-to-MAC address mapping.	**add arp** <ip-address> <mac-address>

Example

The following example adds a static ARP entry to Ethernet interface 1/11/1:

```
cli:172.16.19.10:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# add arp
172.31.1.70 00:10:93:01:ef:d7
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show arp

row count: 1

IP Address       MAC Address        Type
---------------- ------------------ ----------
172.31.1.70      00:10:93:01:ef:d7      static

cli:172.16.19.10:interface:ethernet:csi(1/11/1)#
```

Deleting ARP Entries

Static entries remain in the ARP cache until you manually remove them. You can remove any ARP entry using the following procedure.

Task	Command
1. Enter configuration mode for the selected Ethernet interface.	**interface** <c/s/i>
2. Remove the IP-to-MAC address mapping by specifying the IP address.	**del arp** <ip-address>

Example

```
cli:172.16.19.10:root# interface 1/11/1
mode: interface:ethernet:csi(1/11/1)
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show arp

row count: 1

IP Address       MAC Address        Type
---------------- ------------------ ----------
172.31.1.70      00:10:93:01:ef:d7      static

cli:172.16.19.10:interface:ethernet:csi(1/11/1)# del arp
172.31.1.70
cli:172.16.19.10:interface:ethernet:csi(1/11/1)# show arp

IP Address       MAC Address        Type
---------------- ------------------ ----------

cli:172.16.19.10:interface:ethernet:csi(1/11/1)#
```

Configuring the ARP Timeout

You can set the ARP timeout, which is the timeout in seconds, for dynamic ARP cache entries associated with an interface. When the timeout value is exceeded, the Cuda 12000 flushes out-of-date cache entries from the ARP cache. The cache contains IP addresses and their associated MAC addresses that were stored in response to an ARP reply.

Valid ARP timeout values range from 0 to 604800 seconds. The default is 600 seconds. A value of 0 means that ARP entries will never timeout.

To configure the ARP timeout for an interface, perform the following tasks:

Task	Command
1. Enter configuration mode for the selected interface.	**interface** <c/s/i>
2. Set the ARP timeout.	**arp timeout** <seconds>
3. Verify the ARP timeout.	**show arp timeout**

Note that you can set the ARP timeout to 0 using the **no arp timeout** command.

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# interface 1/11/1
mode:
interface:ethernet:csi(1/11/1)cli:192.168.208.3:interface:ethernet:csi(1/11/
2)# arp timeout 700
cli:192.168.208.3:interface:ethernet:csi(1/11/2)# show arp timeout
ARP Aging                            Enabled
ARP Timeout                              700

cli:192.168.208.3:interface:ethernet:csi(1/11/2)# no arp timeout
cli:192.168.208.3:interface:ethernet:csi(1/11/2)# show arp timeout
ARP Aging                           Disabled
ARP Timeout                                0
```

Clearing the ARP Cache

The **clear arp-cache** command allows you to delete all non-static entries in the ARP cache. If you issue this command from root mode, you delete all non-static ARP entries associated with all interfaces. If you issue this command from interface mode, you delete only the non-static ARP entries associated with the current interface.

Exercise caution when clearing the ARP cache. Clearing the ARP cache can seriously disrupt communications.

To clear the ARP cache for all interfaces on the Cuda, perform these tasks:

Task	Command
1. Enter root mode.	**root**
2. Clear the ARP cache.	**clear arp-cache**

To clear the ARP cache for a specific interface, perform these tasks:

Task	Command
1. Enter interface configuration mode.	**interface** <c/s/i**>**
2. Clear the ARP cache.	**clear arp-cache**

Example

```
cli:192.168.208.3:root# clear arp-cache
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show arp

IP Address       MAC Address        Type
---------------- ------------------ ----------
```

Configuring RIP

This section provides instructions on how to configure Routing Information Protocol (RIP) on an IP interface. Configuring RIP consists of the following tasks:

- Configuring RIP on IP Interfaces
- Disabling RIP on IP Interfaces
- Removing RIP from IP Interfaces

Before you can effectively perform these steps, however, you need to understand certain concepts about RIP.

About RIP

RIP is a distance vector protocol that routers use to build their routing tables dynamically. RIP bases its routing decisions on the distance (number of hops) to a destination. Using RIP, a router requests all or part of the contents of other routers' routing tables. To reply to the requesting router, the other routers send responses that contain the routing table entries.

Your system supports RIP version 2 as defined in RFC 1724. The Cuda 12000 can interoperate in a network of both RIPv1 and RIPv2 routers. A network composed of RIPv1 and RIPv2 routers is useful in supporting the transition from older routers to newer routers supporting RIPv2.

Configuring RIP on IP Interfaces

To exchange RIP routes over an interface, you must configure RIP on that IP interface. After RIP is added to the interface, the Cuda 12000 begins to exchange RIP routes with adjacent RIP routers.

"RIP v1 or RIPv2" mode is enabled on interfaces by default for reception. "RIPv1 compatible" mode is enabled on interfaces by default for transmission.

Before you can configure RIP, you must configure the IP interface on which you want to run RIP. See "Configuring IP Addresses" on page 272 for more information.

Descriptions of the RIP parameters that you configure are listed below.

Table 14-2 RIP Parameters

Parameter	Description
IP Address	IP address of the current interface.
Send Version	Version of RIP packets this router will send on this interface ((RIP v1), (RIP v2), (RIPv1 or RIPv2) or none).
Receive Version	Versions of RIP packets the router will accept on this interface ((RIP v1), (RIP v2), (RIPv1 or RIPv2) or none).
Cost	Cost or metric of the current interface. This cost is added to all routes that are learned from this interface, except the default route if a default cost is specified. Routers use cost values to select the best route to a destination. When multiple routes to a destination exist, routers select the route with the lowest cost.
Authentication ON	Whether authentication is performed on RIP packets received on the interface.
Authentication Type	Method of authenticating RIP packets on this interface. If authentication is performed on the interface, it specifies the type of authentication (md5, simple password, or no authentication) that RIP uses as a security measure to ensure that this interface is exchanging information with proper neighbors.
Send Default Only	Whether the Cuda 12000 advertises only its default route on the interface.
Send Default Also	Whether the Cuda 12000 advertises its default route on the interface (in addition to other routes). The default route is the one that the Cuda 12000 selects when it does not have a specific route to a destination network, subnetwork, or host.

Parameter	Description
Default Cost	Metric that is to be used for the default route entry in RIP updates originated on this interface. A value of zero indicates that no default route should be originated; in this case, a default route via another router may be propagated.
Accept Default	Indicates whether default routes advertised by neighbor routers are accepted on this interface. The default route is the one that routers select when they do not have a specific route to a destination network, subnetwork, or host.
Accept Host Route	Indicates whether host routes advertised by neighbor routers are accepted on this interface. A host route is one in which the destination IP address identifies a specific host, rather than a network or subnet.
Split Horizon	Indicates whether split horizon is used on the interface. Split horizon specifies that if a router learns a route from an update received on the interface, then the router does not advertise that route on updates that it transmits on the interface.
Poisoned Reverse	Indicates whether poison reverse will be used on the interface. Poison Reverse is similar to split horizon, but stronger. Routers do not omit destinations learned from an interface; instead, routers include these destinations in updates, but advertise an infinite cost to reach them. This parameter increases the size of routing updates. In addition, it provides a positive indication that a specific location is not reachable through a router.
Status	Indicates whether RIP is active (enabled) or inactive (disabled) on this interface. When RIP is disabled, RIP packets are not sent or received.

You configure RIP on an IP interface by performing the following tasks:

Task	Command
1. Enter configuration mode for the interface on which you want to enable RIP.	**interface** <c/s/i>
2. Enter the IP address of this interface.	**ip address** <ip-address> <mask>
3. Enable RIP on the current interface.	**ip rip enable**
4. Enter the protocol for outgoing packets that the router will send on this interface. When RIP is enabled for the first time on the interface, the send version defaults to 1 (RIPv1).	**ip rip send-version** {**1** | **2** |**1 2**| **none**}
5. Enter the protocol that the interface uses to learn RIP routes. When RIP is enabled for the first time on the interface, the receive version defaults to 1 2 (RIPv1 or RIPv2).	**ip rip receive-version** {**1**|**2**|**1 2**| **none**}
6. Enter the cost or metric of the current interface. When RIP is enabled for the first time on the interface, the cost defaults to 1.	**ip rip cost** <number>
7. Configure the interface to accept host routes advertised by neighbor routers. When RIP is enabled for the first time on the interface, the interface accepts host routes by default.	**ip rip accept host-route**
8. Configure the interface to accept default routes advertised by neighbor routers. When RIP is enabled for the first time on the interface, the interface accepts default routes.	**ip rip accept default-route**

Task	Command
9. Configure the interface to advertise only the default route. When RIP is enabled for the first time on the interface, the interface does not advertise default routes.	**ip rip send default-only**
10. Configure the interface to advertise the default route in addition to other routes. When RIP is enabled for the first time on the interface, the interface does not advertise default routes.	**ip rip send default-also**
11. Enter the cost or metric that is to be used for the default route entry in RIP updates originated on this interface. A value of zero indicates that no default route should be originated; in this case, a default route via another router may be propagated. When RIP is enabled for the first time on the interface, the default cost is 0.	**ip rip default cost** <number> *Note: Before you can specify a default cost, you must issue the **ip rip send default-only** or **ip rip send default-also** commands.*
12. Configure the interface to implement split horizon. When RIP is enabled for the first time on the interface, split horizon is enabled.	**ip rip split-horizon**
13. Configure the interface to implement poisoned reverse. When RIP is enabled for the first time on the interface, poisoned reverse is enabled.	**ip rip poisoned-reverse**
14. Configure the IP addresses of RIP neighbors. Note that, because RIP can automatically learn the IP addresses of neighbors, you may configure a RIP neighbor that the Cuda 12000 has learned automatically. In this case, the static neighbor entry you create takes precedence over the dynamically created entry.	**ip rip neighbor** <ip-address>

Task	Command
15. Configure the authentication type for the interface. When RIP is enabled for the first time on the interface, no authentication is in effect by default.	**ip rip authentication** {**md5** | **password**}
16. Configure the authentication key for the interface.	**ip rip authentication** {**key-id** <id> **key** <key> | **key** <key>}
17. Verify the RIP configuration on the current interface.	**show ip rip**

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Several of the **ip rip** commands have no forms that enable you to perform the following tasks from within IP address configuration mode:

Task	Command
1. Remove RIP from the interface.	Refer to "Removing RIP from IP Interfaces" on page 297.
2. Disable acceptance of host routes.	**no ip rip accept host-route**
3. Disable acceptance of default routes.	**no ip rip accept default-route**
4. Stop advertising the default route only.	**no ip rip send default-only**
5. Stop advertising the default route in addition to other routes.	**no ip rip send default-also**
6. Disable split horizon.	**no ip rip split-horizon**
7. Disable poisoned reverse.	**no ip rip poisoned-reverse**
8. Delete a static neighbor address.	**no ip rip neighbor** <ip-address>
9. Disable authentication.	**no ip rip authentication**

Example

The following example configures RIP on cable interface 1/1/1:

```
cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# show ip rip

cli:# ip rip enable
cli:# ip rip send-version 2
cli:# ip rip receive-version 2
cli:# ip rip cost 10
cli:# ip rip accept host-route
cli:# ip rip accept default-route
cli:# ip rip poisoned-reverse
cli:# ip rip authentication password
cli:# ip rip authentication key mykey
cli:# show ip rip
IP Address                          201.1.1.1
Send Version                            RIPv2
Receive Version                         RIPv2
Cost                                       10
Authentication ON                        True
Authentication Type          Simple Password
Send Default Only                       False
Send Default Also                       False
Default Cost                                0
Accept Default                           True
Accept Host Route                        True
Split Horizon                            True
Poisoned Reverse                         True
Status                                 Active
```

For more information on the **ip rip** commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide.*

Disabling RIP on IP Interfaces

Use this procedure to disable RIP on an interface:

Task	Command
1. Enter configuration mode for the interface for which you want to disable RIP.	**interface** <c/s/i>
2. Enter the IP address of this interface.	**ip address** <ip-address> <mask>
3. Disable RIP on the interface.	**ip rip disable**

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Removing RIP from IP Interfaces

You remove RIP from an interface by performing the following tasks:

Task	Command
1. Enter configuration mode for the interface from which you want to remove RIP.	**interface** <c/s/i>
2. Enter the IP address of this interface.	**ip address** <ip-address> <mask>
3. Remove RIP from the interface.	**no ip rip**

Configuring OSPF

Open Shortest Path First (OSPF) is a link-state-based interior gateway protocol. With link-state routing protocols, routers maintain a link-state database that contains current information on the state of each communications link in the network topology. This information enables routers to determine the best routes to each destination network within a single autonomous system of networks.

The Cuda 12000 supports OSPF version 2 as defined in RFC 1583. Configuring OSPF consists of the following tasks:

- Configuring OSPF Global Parameters
- Adding OSPF Areas
- Removing OSPF Areas
- Configuring OSPF on IP Interfaces
- Configuring OSPF Virtual Interfaces
- Removing OSPF Virtual Interfaces

Before you can effectively perform these tasks, you need to understand certain concepts about OSPF.

About OSPF

In an OSPF environment, routers exchange link-state advertisements (LSAs) which contain topology information, and they store these LSAs in link-state databases. OSPF ensures that all routers have identical databases and the same topology information.

Link-state databases include router addresses, links and their associated costs, and network addresses. Routers use the link-state database and Dijkstra's algorithm (algorithm used to calculate best routes) to choose the best route.

Link-state database size increases in proportion to network size. This causes several problems. Demand for router resources, such as memory and CPU time, can increase significantly. It also takes longer to calculate link costs for more links and generate large routing tables that large networks require.

OSPF Areas

To address large link-state databases, OSPF employs *areas,* which are groups of OSPF routers that exchange topology information. Designated routers only send LSAs to routers in the same area. If an autonomous system (AS) has one area, all routers in the AS receive LSAs; however, if the AS consists of many areas, LSAs are only sent to the appropriate areas, which minimizes traffic and the link-state database size. Using areas, the AS works much like a group of small networks.

Figure 14-1 Illustrates the concept of areas.

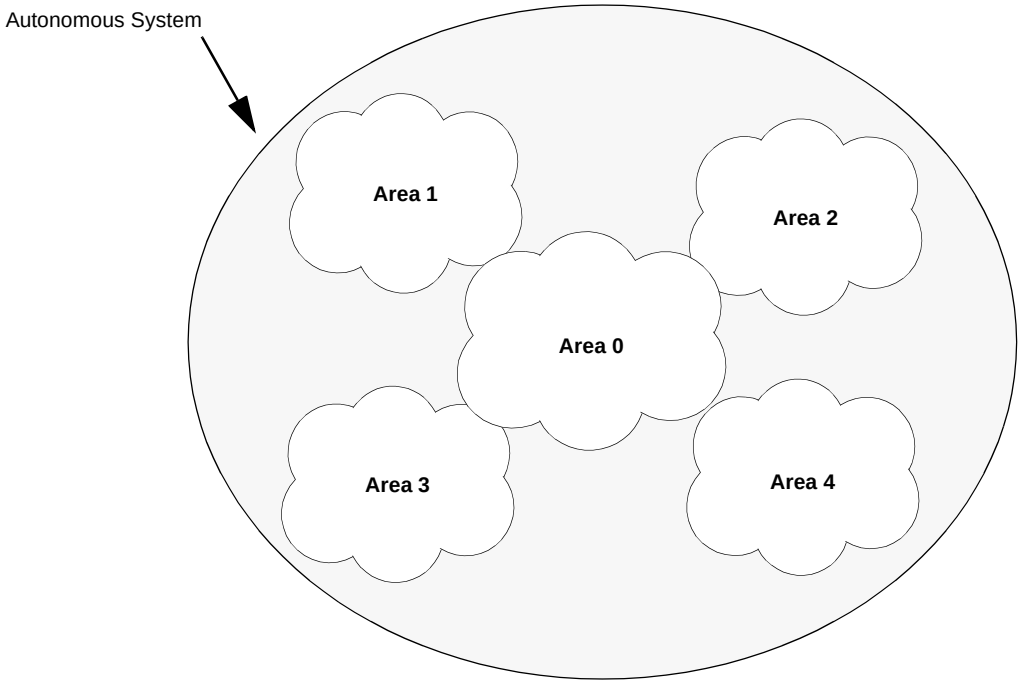


Figure 14-1 OSPF Areas

In this example, the AS consists of five areas. Each area represents smaller networks within the AS, and maintains separate link-state databases. Area 0 is the backbone, connecting all areas within the AS.

OSPF requires the backbone to be contiguous to all areas in the AS. All other non-backbone areas (areas other than area 0) must have a connection to the backbone area. Non-backbone areas that are not contiguous to the backbone can use *virtual interfaces* to connect to the backbone.

OSPF Routers

OSPF performs routing within an area and routing between areas. Different categories of routers perform these two types of routing.

Autonomous system boundary routers (ASBRs) connect an OSPF routing domain to another routing domain (such as RIP).

Area border routers (ABRs) are linked to more than one area, or are linked between an area and backbone. ABRs maintain separate link-state databases of each area in which they reside.

Internal routers are directly connected within the same area, as well as routers with interfaces connected only to the backbone.

Figure 14-2 Shows an example of OSPF routing and router classifications.

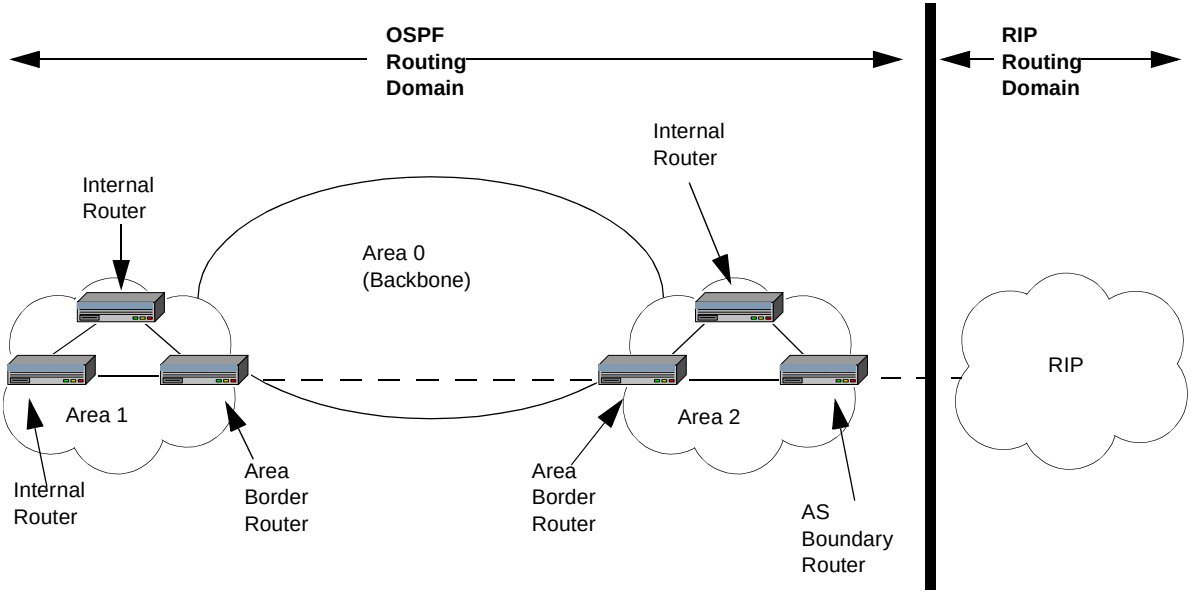


Figure 14-2 OSPF Router and Routing Classifications

OSPF Configuration Task Overview

Configuration of the Cuda 12000 within an OSPF network includes:

- Configuring OSPF Global Parameters
- Adding OSPF Areas
- Removing OSPF Areas
- Configuring OSPF on IP Interfaces
- Removing OSPF from IP Interfaces
- Configuring OSPF Virtual Interfaces
- Removing OSPF Virtual Interfaces
- Configuring OSPF Neighbor Traps

Configuring OSPF Global Parameters

OSPF Global Parameters provide information about this router to other OSPF routers in the autonomous system. These global parameters include:

Table 14-3 OSPF Global Parameters

Parameter	Description
OSPF Router ID	The OSPF router ID uniquely identifies this router to other routers in the autonomous system. It is recommended that you use one of the system's IP addresses or define an IP address that is unique to the network.
Autonomous System Boundary Router (ASBR)	Configures this router as an ASBR.
OSPF Administration State	Enables or disables the OSPF protocol on a system-wide basis. If enabled, this router participates in the OSPF network; otherwise the system does not participate.
OSPF Neighbor State and OSPF Virtual Neighbor State Traps	Enables or disables sending of OSPF neighbor state and OSPF virtual neighbor state traps. Refer to "Configuring OSPF Neighbor Traps" on page 318 for more information.

OSPF global parameters are configured within router ospf mode. You configure OSPF global parameters by performing the following tasks:

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Define the OSPF router ID for this system.	**router-id** <ip address> *Note that you cannot set the router ID to 0.0.0.0 or 255.255.255.255.*
3. Configure this router as an ASBR.	**asbr** *Note that ASBR is disabled by default. If you configure this router as an ASBR, and then later decide to disable ASBR, issue this command:* **no asbr**
4. Display OSPF global parameters to verify configuration. You can display OSPF global parameters from within any mode.	**show ospf**

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Example

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# router-id 201.1.1.1
cli:172.16.19.10:router:ospf# asbr
cli:172.16.19.10:router:ospf# show ospf
Admin Status                          Enabled
TOS Support                             False
Router Id                            201.1.1.1
ABR Status                              False
ASBR Status                             True

Report ospf-nbr-state                Disabled
Report ospf-virt-nbr-state           Disabled
```

Adding OSPF Areas

You can divide an AS into smaller, more manageable sub-divisions or areas. This reduces the amount of routing information that must travel through the network and serves to reduce the size of each router's routing database.

In order for the Cuda 12000 to support OSPF, you must add at least one area. Typically, the Cuda 12000 will have a direct connection to the OSPF backbone, in which case you must add area 0.0.0.0. If the Cuda 12000 does not have a direct connection to the backbone, you must configure an OSPF virtual interface to the backbone. Refer to "Configuring OSPF Virtual Interfaces" on page 313 for more information on configuring virtual interfaces.

When you add an area on a Cuda 12000 acting as an ABR, you can implement area range summarization (also called "route summarization") for the area. With area range summarization, a single router summary is advertised to other areas, thus reducing routing traffic and saving LSA database space.

You add OSPF areas by performing the following tasks:

Task	Command
1. Enter router ospf mode.	**router ospf**
2. Define the area ID. The area ID is specified in the form of an IP address.	**ospf area** <area-id>
3. To configure the area as a stub area: If you configure the router as a stub area, it does not flood external link advertisements into the area. Instead, it advertises a single default external route into the area. This conserves LSA database space that would otherwise be used to store external link state advertisements.	**ospf area** <area-id> **stub**

Task	Command
4. Configure the default cost only if this is an ABR connected to a stub area. The default cost is the metric assigned to the summary default route injected into the area by this router.	**ospf area** <area-id> **default-cost** <cost value>
5. If you want to summarize routes injected into the area, use this command to define the area range.	**ospf area** <area-id> **range** <ip address> <mask> [**advertise-matching**]
6. Configure authentication for this area.	**ospf area** <area-id> **authentication** {**md5** | **password**} Specify "md5" to configure MD5 password authentication; "password" to configure simple password authentication.
7. Enable or disable this OSPF area.	**ospf area** <area-id> {**enable** | **disable**}
8. Display the OSPF area to verify configuration.	**show ospf area** <area-id>

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

The no forms of the **ospf** command enable you to perform the following tasks from within router ospf mode:

Task	Command
1. Remove an OSPF area.	Refer to "Removing OSPF Areas" on page 305.
2. Remove a summary range.	**no ospf area** <area-id> **range** <ip address> <mask> [**advertise-matching**]
3. Disable authentication for the area.	**no ospf area** <area-id> **authentication**

Example

```
cli# router ospf
mode: router:ospf
cli# ospf area 0.0.0.1
cli# ospf area 0.0.0.1 authentication md5
cli# ospf area 0.0.0.1 stub
cli# ospf area 0.0.0.1 default-cost 10
cli#
```

Removing OSPF Areas

You remove a specified OSPF area by performing the following tasks:

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Display OSPF areas.	**show ospf area**
3. Remove the OSPF area.	**no ospf area** <area id>

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Example

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# show ospf area

row count: 1

Area Id          Auth Type ImpAsExt SPF  ABR   ASBR  LSA   LSA    Area    Area
                                        Count Count Count Cksum  Summary Type
---------------- --------- -------- ---- ----- ----- ----- ------ ------- -----
2.2.2.2                MD5   Extern    1     0     0     1  56944    Send  Norm

cli:172.16.19.10:router:ospf# no ospf area 2.2.2.2
cli:172.16.19.10:router:ospf# show ospf area


Area Id          Auth Type ImpAsExt SPF  ABR   ASBR  LSA   LSA    Area    Area
                                        Count Count Count Cksum  Summary Type
---------------- --------- -------- ---- ----- ----- ----- ------ ------- -----

cli:172.16.19.10:router:ospf#
```

Configuring OSPF on IP Interfaces

You configure OSPF on IP interfaces using the **ip ospf** command within interface configuration mode. To configure OSPF on an IP interface, you must:

- Assign an area ID to the IP interface
- Configure OSPF parameters for the IP interface

Assigning an Area ID to the IP Interface

Before you can configure any OSPF parameters on an IP interface, you must assign the area ID to the IP interface. Consider these guidelines:

- The IP addresses of interfaces that connect OSPF neighbors to each other must have the same network number (or the same network number and subnetwork number if subnetting is used), and the same mask. Only the host portion of the IP address can be unique.
- To form OSPF adjacencies, all OSPF neighbors must have a matching area ID configured on the IP interfaces that connect them to each other. For example, suppose a Cuda 12000 and two other nodes are OSPF neighbors. On the IP interfaces that connect all three systems to each other, you must configure an identical area ID.
- To form OSPF adjacencies, the area ID that you assign to OSPF neighbor interfaces must identify the same area type. For example, if you are assigning a stub area, all the OSPF neighbors must be in agreement.

To assign an area ID to an IP interface, perform the following tasks:

Task	Command
1. Create the IP interface.	Refer to "Configuring IP Addresses" on page 272.
2. Enter IP address configuration mode for the interface.	**ip address** <ip-address> <mask>
3. Assign an area ID to the interface.	**ip ospf area-id** <area-id>

Example

The following example enters IP address configuration mode for interface 1/1/1 and assigns the interface an area ID of 1.1.1.1.

```
cli:172.16.19.10:root# router ospf
mode: router:ospf
cli:172.16.19.10:router:ospf# ospf area 1.1.1.1
cli:172.16.19.10:router:ospf# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip address 172.16.9.8
255.255.255.0
cli:172.16.19.10:interface:cable:csi(1/1/1):ip-address(172.16.9.8)# ip ospf
area-id 1.1.1.1
cli:172.16.19.10:interface:cable:csi(1/1/1):ip-address(172.16.9.8)# show ip
ospf
IP Address                        172.16.9.8
Area ID                              1.1.1.1
Type                                   Bcast
Priority                                   5
Transit Delay                              1
Retrans Int                                5
Hello Int                                 10
Dead Int                                  40
Poll Int                                   0
Admin Stat                           Enabled
Status                                Active
Auth Type                               None

Auth Key Id                                0
Cost                                       1
```

Configuring OSPF Parameters for an IP Interface

After you have entered IP address configuration mode and assigned an area ID to the IP interface, you can then configure OSPF parameters on the interface.

To do so, perform the following tasks within IP address configuration mode:

Task	Command
1. Configure the cost for this OSPF interface.	**ip ospf cost** <number>
2. Configure the dead-interval for this OSPF interface, in seconds. When OSPF is initially enabled on the interface, the dead interval defaults to 40 seconds. The Cuda 12000 and all of its OSPF neighbors on the IP interface you are configuring must have matching dead interval values. Otherwise, the neighbors cannot form adjacencies.	**ip ospf dead-interval** <seconds>
3. Configure the Hello interval for the current OSPF interface, in seconds. When OSPF is initially enabled on the interface, the hello interval defaults to 10 seconds. The Cuda 12000 and all of its OSPF neighbors on the IP interface you are configuring must have matching hello interval values. Otherwise, the neighbors cannot form adjacencies.	**ip ospf hello-interval** <seconds>

Task	Command
4. Configure the interface priority. This number identifies the priority of the Cuda 12000 relative to other OSPF routers on the current interface. The number is used to elect the designated and backup designated routers. The router with the highest priority is considered the designated router. A value of 0 indicates that the router is not eligible to be the designated or backup designated router. If all routers have the same priority, the router ID is used to determine the designated router. When OSPF is initially enabled on the interface, the priority defaults to 5.	**ip ospf priority** <number>
5. Configure the retransmit interval, in seconds. When OSPF is initially enabled on the interface, the retransmit interval defaults to 5 seconds. While it is common practice to configure all OSPF neighbors with the same retransmit interval value, it is not required.	**ip ospf retransmit-interval** <seconds>
6. Configure the transit-delay, in seconds. When OSPF is initially enabled on the interface, the transit delay defaults to 1 second. While it is common practice to configure all OSPF neighbors with the same transit delay value, it is not required.	**ip ospf transit-delay** <seconds>

Task	Command
7. Configure the authentication type. Authentication is optional. If you choose to implement authentication: ■ The Cuda 12000 and all of its OSPF neighbors on the interface you are configuring must implement the same type of authentication. ■ The type of authentication you specify must match the authentication type specified for the area. See "Adding OSPF Areas" on page 303 for more information on specifying authentication types for areas.	**ip ospf authentication** {**md5 | password**} *Note that you specify "md5" to configure MD5 password authentication; "password" to configure simple password authentication.*
8. Configure the authentication key. Authentication is optional. If you choose to implement authentication, the Cuda 12000 and all of its OSPF neighbors on the IP interface you are configuring must use the same password (for simple password) or key and corresponding key ID (for MD5).	**ip ospf authentication** {**key-id** <id> **key** <key> **| key** <key>} *Use the key-id <id> key <key> argument with MD5; use key <key> with simple password.*
9. Verify the OSPF interface configuration.	**show ip ospf**

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

A couple of the **ip ospf** commands have no forms that enable you to perform the following tasks from within IP address configuration mode:

Task	Command
1. Remove OSPF from the interface.	Refer to "Removing OSPF from IP Interfaces" on page 312.
2. Disable authentication.	**no ip ospf authentication**

Example

The following example configures OSPF parameters on IP interface 201.1.1.1:

```
cli:# router ospf
mode: router:ospf
cli:# show ospf area

row count: 2

Area Id          Auth Type ImpAsExt SPF  ABR   ASBR  LSA   LSA    Area    Area
                                        Count Count Count Cksum  Summary Type
---------------- --------- -------- ---- ----- ----- ----- ------ ------- -----
0.0.0.0               None   Extern   32     0     0     1  26284    Send  Norm
0.0.0.1               None   Extern    8     0     0     1  45702    Send  Norm

cli:# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:# ip address 201.1.1.1 255.255.255.0
cli:# ip ospf area 0.0.0.1
cli:# ip ospf cost 500
cli:# ip ospf dead-interval 500
cli:# ip ospf hello-interval 500
cli:# ip ospf priority 100
cli:# ip ospf retransmit-interval 500
cli:# ip ospf transit-delay 500
cli:# ip ospf authentication password
cli:# ip ospf authentication key mysecret
cli:# show ip ospf
IP Address                          201.1.1.1
Area ID                               0.0.0.1
Type                                    Bcast
Priority                                  100
Transit Delay                             500
Retrans Int                               500
Hello Int                                 500
Dead Int                                  500
Poll Int                                    0
Admin Stat                            Enabled
Status                                 Active
Auth Type                     Simple Password

Auth Key Id                                 0
Cost                                      500
```

Removing OSPF from IP Interfaces

You remove OSPF from an interface by performing the following tasks:

Task	Command
1. Enter configuration mode for the interface from which you want to remove OSPF.	**interface** <c/s/i>
2. Enter the IP address of this interface.	**ip address** <ip-address> <mask>
3. Remove OSPF from the interface.	**no ip ospf area-id** <area-id>

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Configuring OSPF Virtual Interfaces

OSPF requires that all areas be attached to the OSPF backbone area (area 0.0.0.0). However, you may encounter situations in which you cannot connect an OSPF area directly to the backbone. If your Cuda 12000 is an area border router between one area that is physically connected to the OSPF backbone and one area that is not, you can create a virtual interface on your Cuda 12000 to connect the non-contiguous area to the OSPF backbone.

OSPF uses virtual interfaces for the following purposes:

- Connecting areas that are not physically connected to the backbone.
- Repair the backbone if a break in backbone continuity occurs.

Before you configure a virtual interface on your Cuda 12000, make sure that:

- The OSPF backbone (area 0) is configured.
- You know the area ID of the transit area, which is the area that connects the non-contiguous area to the backbone.
- You know the router ID of the OSPF neighbor that acts as the ABR between the transit area and the backbone.

For example, if your Cuda 12000 is an ABR between an area that is contiguous to the backbone (area 0.0.0.1) and an area that is not contiguous to the backbone (area 0.0.0.2), as shown in Figure 14-3.

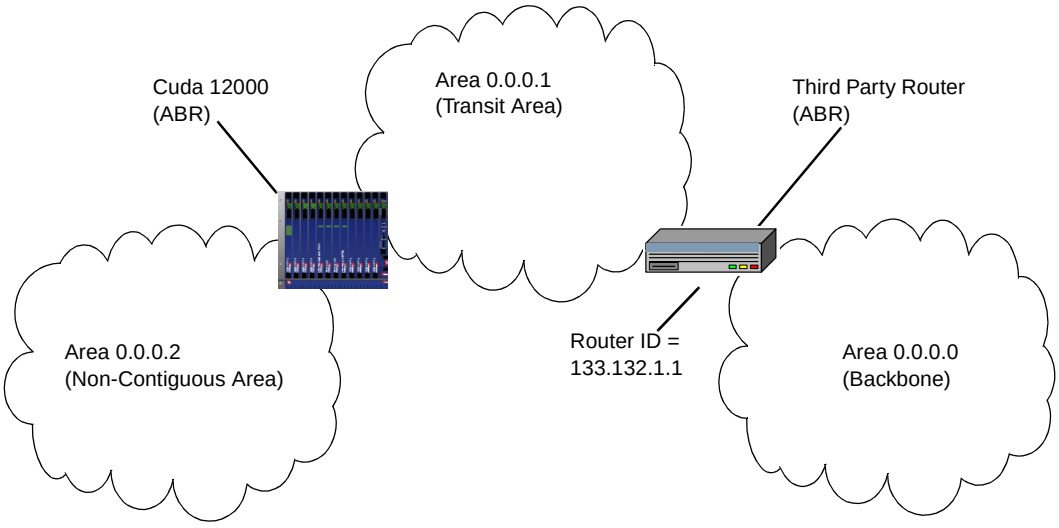


Figure 14-3 Sample Non-Contiguous Area

To connect area 0.0.0.2 to the backbone, you would have to create a virtual interface on the Cuda 12000. You would specify 0.0.0.1 for the transit area and 133.132.1.1 for the neighbor router ID.

Use this procedure to configure a virtual interface:

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Configure the transit area and the router ID of the neighbor for this virtual interface.	**ospf-vi** <transit-area-id> <neighbor-router-id>
3. Configure the dead-interval, in seconds. OSPF neighbors connected by a virtual interface must have matching dead interval values in their respective virtual interface configurations.	**ospf-vi** <transit-area-id> <neighbor-router-id> **dead-interval** <seconds>
4. Configure the Hello interval, in seconds, in seconds. OSPF neighbors connected by a virtual interface must have matching hello interval values in their respective virtual interface configurations.	**ospf-vi** <transit-area-id> <neighbor-router-id> **hello-interval** <seconds>
5. Configure the retransmit interval, in seconds. While it is common practice to configure OSPF neighbors connected by a virtual interface with matching retransmit interval values, it is not required.	**ospf-vi** <transit-area-id> <neighbor-router-id> **retransmit-interval** <seconds>
6. Configure the transit-delay, in seconds. While it is common practice to configure OSPF neighbors connected by a virtual interface with matching transit delay values, it is not required.	**ospf-vi** <transit-area-id> <neighbor-router-id> **transit-delay** <seconds>

Task	Command
7. Configure the authentication type. Authentication is optional. If you choose to implement authentication, the OSPF neighbors on the virtual interface must all implement the same authentication type.	**ospf-vi** <transit-area-id> <neighbor-router-id> **authentication** {**md5 | password**} *Note that:* ■ *Specify "md5" to configure MD5 password authentication; "password" to configure simple password authentication.* ■ *Use this command to disable authentication for the virtual interface*: **no ospf-vi** <transit-area-id> <neighbor-router-id> **authentication**
8. Configure the authentication key. Authentication is optional. If you choose to implement authentication, all OSPF neighbors on the virtual interface must use the same password (for simple password) or key and corresponding key ID (for MD5).	**ospf-vi** <transit-area-id> <neighbor-router-id> **authentication {key-id** <id> **key** <key> **| key** <key>} *Use the key-id <id> key <key> argument with MD5; use the key <key> argument with password.*
9. Verify the OSPF virtual interface configuration.	**show ospf interface virtual**
10. Verify OSPF virtual neighbors.	**show ospf neighbor virtual**

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Example

The following example illustrates how to create a virtual interface that links a non-contiguous area to the backbone via a transit area of 0.0.0.1 and an ABR with a router ID of 133.132.1.1:

```
cli:192.168.208.3:root# router ospf
mode: router:ospf
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1 dead-interval 500
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1 hello-interval 400
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1 retransmit-interval
350
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1 transit-delay 150
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1 authentication md5
cli:192.168.208.3:router:ospf# ospf-vi 0.0.0.1 133.132.1.1 authentication
key-id 44231 key ertiddkdkk
cli:192.168.208.3:router:ospf# show ospf interface virtual

row count: 1

Transit       Neighbor       State Transit Retrans Hello Dead Auth Event Status
Area                               Delay   Int     Int   Int  Type Count
------------- -------------- ----- ------- ------- ----- ---- ---- ----- ------
0.0.0.1       133.132.1.1    up        150     350   400  500 MD5       0 Active
```

Removing OSPF Virtual Interfaces

Use this procedure to remove a virtual interface:

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Display virtual interfaces.	**show ospf interface virtual**
3. Remove the virtual interface.	**no ospf-vi** <transit-area-id> <neighbor-router-id>

For more information on these commands, refer to the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Configuring OSPF Neighbor Traps

You can enable the sending of the OSPF neighbor state trap and the OSPF virtual neighbor state trap. These traps report changes in OSPF neighbor state or virtual neighbor state.

To enable OSPF neighbor traps, perform the following tasks:

Task	Command
1. Enter router-ospf mode.	**router ospf**
2. Enable the OSPF neighbor state trap.	**report ospf-nbr-state** *Note: To disable the trap, issue this command:* **no report ospf-nbr-state**
3. Enable the OSPF virtual neighbor state trap.	**report ospf-virt-nbr-state** *Note: To disable the trap, issue this command:* **no report ospf-virt-nbr-state**
4. Use the **event-config reporting** command to configure how different classes of events are reported. The OSPF neighbor state and OSPF virtual neighbor state traps are in the Notify class.	Refer to Chapter 10 "Managing System Events" for more information.
5. Use the **snmp-server host** command to indicate that you want the ospf-nbr-state and ospf-virt-nbr-state notification types to be sent to the specified SNMP management station. *Note: You can skip this step if you do not want changes in the OSPF neighbor state or OSPF virtual neighbor state to be reported as an SNMP trap.*	Refer to Chapter 9 "Simple Network Management Protocol (SNMP)" for more information.

Example

```
cli:192.168.220.230:router:ospf# report ospf-nbr-state
cli:192.168.220.230:router:ospf# report ospf-virt-nbr-state
cli:192.168.220.230:router:ospf# show ospf
Admin Status                          Enabled
TOS Support                             False
Router Id                           201.1.1.1
ABR Status                              False
ASBR Status                             False

Report ospf-nbr-state                 Enabled
Report ospf-virt-nbr-state            Enabled
```

For **snmp-server host** command examples, refer to Chapter 9 "Simple Network Management Protocol (SNMP)" for more information.

For **event-config reporting** command examples, refer to Chapter 10 "Managing System Events" for more information.

Configuring IP Source Routing

IP source routing allows you to configure the default route a packet should take based on the source IP address of the packet. Configuring IP source routing includes the following tasks:

- Adding IP Source Routes
- Displaying IP Source Routes
- Removing IP Source Routes

In addition, a sample source routing configuration is provided.

Before you can effectively perform source routing configuration tasks, you need to understand some concepts behind IP source routing.

About IP Source Routing

Source routing allows you to configure a different default route for each IP network or host. Specifically, source routing allows you to define the default route *(next hop gateway)* to which a packet containing a particular source IP address should be forwarded in the event that a local route to the destination does not exist. This feature is called source routing because the route is determined by the source of the message.

When an IP packet is received on an interface:

- The interface performs a normal destination-based route lookup.
- If the system finds no route, or if a default route exists for the destination IP address, it then checks the source routing entries defined on the interface.
- If the system finds a source routing entry defined for the source address, the packet is forwarded to the next hop gateway associated with the source address. Otherwise, the default route defined in the routing table is used.

This logic enables local routes to take precedence over both default routes and source routing entries; but enables source routing entries to take precedence over default routes. This allows local servers, such as mail servers, provisioning servers, and web caching servers to be used first. But if no local routes exist, you can define the next hop gateway based on the source address of the host.

Source routing is configured on an interface using the **ip source-route** command. This command allows you to define the following source routing criteria:

- **Source IP address to match**: An IP address and network mask combination allows you to define the source route to match as a network, or scope it down to a specific host.
- **Next Hop Gateway**: The IP address to which the system must forward any matching IP datagrams. Note that you must enter a valid next hop destination.

Adding IP Source Routes

Source routes are added on a per-interface basis. You add a source route entry on a particular interface by performing the following tasks:

Task	Command
1. Enter interface mode.	**interface** <c/s/i>
2. Add a source route.	**ip source-route** <ip address> <mask> <next hop gateway>

Examples

The following example configures interface 1/1/1 to forward any packets received from the 172.16.19.0 network to the router at 172.20.19.4:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip source-route
172.16.19.0 255.255.255.0 172.20.19.4
```

Similarly, the following example configures interface 1/1/1 to forward any packets received from host 172.16.19.4 to a next-hop destination of 171.16.19.50:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip source-route
172.16.19.4 255.255.255.255 172.20.19.50
```

Displaying IP Source Routes

You display the source routing entries configured on a particular interface by performing the following tasks:

Task	Command
1. Enter interface mode.	**interface** <c/s/i>
2. Display a source route entry.	**show ip interface source-route**

Example

The following example displays all source route entries configured on interface 1/1/1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip interface
source-route

Address          Mask             NextHop
---------------- ---------------- ----------------
     172.16.19.0    255.255.255.0      172.20.19.4
     173.16.19.0    255.255.255.0      172.20.19.4
    173.200.19.0    255.255.255.0      172.20.19.4

row count: 3
```

Removing IP Source Routes

To remove an IP source route entry from a particular interface, perform the following tasks:

Task	Command
1. Enter interface mode.	**interface** <c/s/i>
2. Remove a source route entry.	**no ip source-route** <ip address> <mask> <next hop gateway>

The following example configures interface 1/1/1 to forward any packets received from the 172.16.19.0 network to the router at 172.20.19.4:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# no ip source-route
172.16.19.0 255.255.255.0 172.20.19.4
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip interface
source-route

Address          Mask             NextHop
---------------- ---------------- ----------------
     172.16.19.0    255.255.255.0      172.20.19.4

row count: 1
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip interface
source-route

Address          Mask             NextHop
---------------- ---------------- ----------------

row count: 0
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Source Routing Configuration Example

For example, if 3 users are connected to a single DOCSIS module on interface 1/1/1; each user would belong to a different internet service provider, ISP-1, ISP-2, ISP-3, as described in the following table:

Customer Host	ISP	ISP's Router
172.16.19.2	ISP-1	209.16.19.2
172.16.19.5	ISP-2	172.16.20.4
172.16.19.9	ISP-3	172.19.34.5

To configure each customer to use a default route that points to their ISP's router, you would configure the following source routing entries:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip source-route 172.16.19.2
255.255.255.255 209.16.19.2
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip source-route 172.16.19.5
255.255.255.255 172.16.20.4
cli:172.16.19.10:interface:cable:csi(1/1/1)# ip source-route 172.16.19.9
255.255.255.255 172.19.34.5
cli:172.16.19.10:interface:cable:csi(1/1/1)# show ip interface source-route

Address          Mask             NextHop
---------------- ---------------- ----------------
     172.16.19.2  255.255.255.255      209.16.19.2
     172.16.19.5  255.255.255.255      172.16.20.4
     172.16.19.9  255.255.255.255      172.19.34.5

row count: 3
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

15 IP PACKET FILTERING

This chapter covers IP packet filtering on the Cuda 12000 and includes the following sections:

Note that IP packet filtering is only supported on cable interfaces.

About IP Packet Filtering

The Cuda 12000 supports packet filtering in the form of access lists. Access lists allow you to restrict and control IP packet flow over specified cable interfaces. This control of IP packet transmission restricts network access from specified users, devices, and applications.

IP packet filtering involves the following steps:

1. Create access lists that define the IP packet filtering criteria using the **access-list** command as described in "Creating Access Lists" on page 331.
2. Apply the access lists to specified interfaces using the **access-class** command as described in "Applying Access Lists to Interfaces" on page 336.
3. Enable IP packet filtering on the specified interface as described in "Enabling and Disabling IP Packet Filtering," next.

Enabling and Disabling IP Packet Filtering

Whenever you apply an access-list to an interface using the **access-class** command IP filtering is automatically enabled.

You can disable IP filtering so that all packets are permitted to cross the interface. You must disable access lists manually; IP filtering is not automatically disabled when access lists are removed.

For each interface, you can enable filtering of incoming packets, as well as outgoing packets. To do so, perform the following tasks within interface <c/s/i> mode:

Task	Command
1. Enable packet filtering on the current interface.	**ip filter** {**in** | **out**} **enable**
2. Disable packet filtering on the current interface.	**ip filter** {**in** | **out**} **disable**

Understanding Access Lists

Access lists are sequential groupings of permit and deny rules. These rules enable you to permit or deny packets from crossing specified interfaces. An access list is comprised of both match criteria and actions to take upon finding a match.

Match criteria can include:

- Source IP address
- Destination IP address
- Source TCP/UDP port
- Destination TCP/UDP port
- TCP Sync Flag
- TCP Establish State
- IP Type of Service (TOS)

Actions that can be taken against matching packets include:

- Permit
- Deny
- Change IP TOS

Access lists are pooled and indexed on a system-wide basis. As such, you can create access-lists in either root mode, or interface configuration mode. Access lists are then only used by an interface when you enable IP filtering on the interface and apply the predefined access-lists to the interface using the **access-class** command. Each access-list is identified by a list number that you define when creating the list.

You cannot modify an existing access list, which means that if you want to change an access list, you must delete it and then recreate it with the same name.

Creating Access Lists

Access lists are comprised of rules that are sequenced according to assigned rule numbers. These rules are created and assigned to access lists using the **access-list** command. Packets are matched against the lowest numbered rules first.

Each rule defines a permit or deny action which determines whether the packet is accepted or permitted when matched. Note that access lists include an implicit deny command at the end. This means that an IP filter-enabled interface rejects (drops) packets for which no match is found.

Figure 15-1 shows a logical representation of an access list:

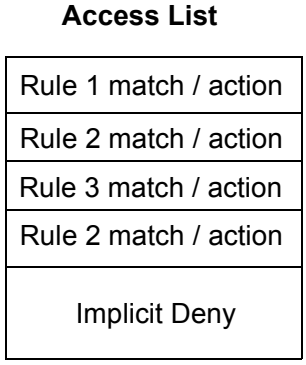


Figure 15-1 Access List

You can use access lists to filter the following protocols:

- Internet Protocol (IP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Note that when masking network addresses, 0 indicates "care" bits; 1 indicates "don't care." For example, a class C network would be masked as 0.0.0.255.

Creating IP Access Lists

To create an IP access list, perform the following task in either root mode or interface configuration mode:

Task	Command
Create an IP access list.	**access-list** <list number> {**deny** | **permit**} <rule number> **ip** {<source IP address> <source IP mask> | host <ip address> | **any**} {<destination IP address> <destination IP mask> | **host** <destination ip address> | **any**} [**tos** <tos> <tos mask>] [**change-tos** <tos>]

For example, the following access list denies IP packets with the source address of 172.16.19.200:

```
cli:172.16.19.10:root# access-list 4 deny 10 ip 172.16.19.200
0.0.0.0 any
```

Creating TCP Access Lists

To create a TCP access list, perform the following task in either root mode or interface configuration mode:

Task	Command
Create an TCP access list.	**access-list** <list number> {**deny** | **permit**} <rule number> **tcp** [<source IP address> <source IP mask> | **host** <ip address> | **any**} {<destination IP address> <destination IP mask> | **host** <destination ip address> | **any**} {<ip address> <IP mask> | **host** <ip address> | <operator> <port> [<port>] | **any**}] {< IP address> < mask > | **any** | **host** <ip address> [<operator> <port> [<port>]] **established**] [**tos** <tos> <tos mask>] [**change-tos** <tos>]

For example, the following access list permits TCP traffic on port 23 (Telnet) from host 172.16.19.200 to any IP address destination.

```
cli:172.16.19.10:root# access-list 5 permit 1 tcp 172.16.19.200
0.0.0.0 any eq 23
```

Creating UDP Access Lists

To create a UDP access list, perform the following task in either root mode or interface configuration mode:

Task	Command
Create an UDP access list.	**access-list** <list number> {**deny** | **permit**} <rule number> **udp** [<source IP address> <source IP mask> | **host** <ip address> | **any** } {<destination IP address> <destination IP mask> | **host** <destination ip address> | **any**} {<ip address> <IP mask> | **host** <ip address> | <operator> <port> [<port>] | **any**}] {< IP address> < mask > | **any** | **host** <ip address> [<operator> <port> [<port>]] [**established**] [**tos** <tos> <tos mask>] [**change-tos** <tos>]

The following example will prevent UDP traffic from host 172.16.10.200 from traveling over port 50:

```
cli:172.16.19.10:root# access-list 6 deny 1 udp 172.16.19.200
0.0.0.0 any eq 50
```

The following table provides a quick reference to **access list** command arguments. For more information, see the *Cuda 12000 IP Access Switch CLI Reference Guide*.

Table 15-1 Access List Command Arguments

Argument	Description
list number	Index number that identifies this list. Valid range: 1–65535.
rule number	Number identifying the precedence of this access list. Smaller rule numbers result in greater precedence. This means that an access list with a lesser rule number is applied against the interface first.
source ip address	IP address seen in the source IP address field of the protocol header. A value of **any** acts as a wildcard.
source ip mask	Source IP address network mask, if you specified a specified address.
destination ip address	IP address seen in the destination IP address field of the protocol header. A value of **any** acts as a wildcard.
destination ip mask	Destination IP address network mask, if you specified a specific address.
host	Host address if filtering on a specific IP host.
operator	Operand used to compare source and destination ports. You can use the following operands: ■ **lt** (less than) ■ **gt** (greater than) ■ **eq** (equal) ■ **range** (inclusive range)
port	TCP or UDP port number. Valid range: 0–65535.
established	For TCP protocol only. Indicates an established TCP connection. Match occurs when the ACK or RST bits of a TCP datagram are set.
tos	Type of Service level identified in the IP packet header. Valid Range 0 – 15.
tos-mask	Type of Service mask.

Displaying Access Lists

To display the contents of a specified access list, perform the following task within root or interface <c/s/i> mode:

Task	Command
1. Display all access lists within the system.	**show access-list**
2. Display a specified access list.	**show access-list** <list number>

Deleting Access Lists

To delete an access list, perform the following task within root or interface <c/s/i> mode:

Task	Command
1. To delete a specified access list.	**no access-list** *<list number>*
2. To remove all access lists from the system.	**no access-list all**

Note that when filtering is enabled with no access lists applied to the interface, the interface permits all traffic to pass.

Applying Access Lists to Interfaces

After you create an access list, you can apply it to one or more interfaces to filter traffic. Filters can be applied to either outbound or inbound interfaces or both.

Note that filtering is enabled automatically when you apply an access list to an interface.

You apply access lists to a specific interface by using the **access-class** command. To do so, perform the following task within interface <c/s/i> mode:

Task	Command
Apply filters to the current interface to restrict incoming or outgoing traffic.	**access-class** <list number> {**in | out**} **priority** <priority number>

The following example applies access list 1 to the inbound interface and access list 2 to the outbound interface of cable interface 1/1/1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-list

List  Ac Rule Prot IP Source       Mask Source     Start End   Estab TOS ChTOS
                   IP Dest         Mask Dest       Port  Port  Sync  Msk
----- -- ---- ---- --------------- --------------- ----- ----- ----- --- -----
    1 DE    1   ip   172.16.19.200         0.0.0.0             False   0
                          0.0.0.0 255.255.255.255             False   0
..... .. .... .... ............... ............... ..... ..... ..... ... .....
    2 PE    1  tcp   172.16.19.200         0.0.0.0     0 65535 False   0
                          0.0.0.0 255.255.255.255     0 65535 False   0
..... .. .... .... ............... ............... ..... ..... ..... ... .....

row count: 2
cli:172.16.19.10:interface:cable:csi(1/1/1)# access-class 1 in priority 1
cli:172.16.19.10:interface:cable:csi(1/1/1)# access-class 2 out priority 2
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-class

Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------
     1        in        1      1
     2       out        2      1

row count: 2
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Just as the rule number determines the sequence of rule examination within an access list, priority specifies the order of access list examination within the access class that you apply to an inbound or outbound interface.

Figure 15-2 shows a logical representation of an access class for an inbound or outbound interface.

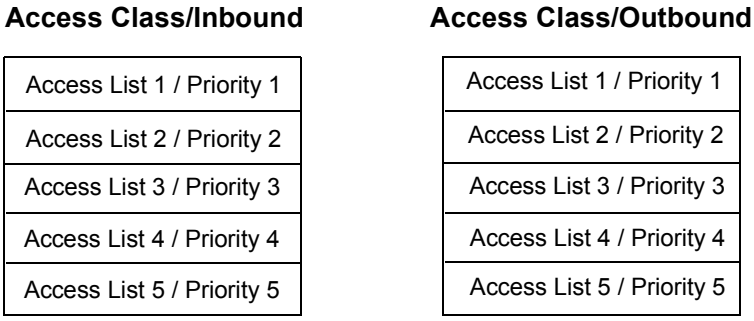


Figure 15-2 Access Class

Displaying Access Classes

To display the access lists applied to an interface, display the access class for the interface. To do so, perform the following task within interface <c/s/i> mode:

Task	Command
Display the access class configuration for a specified interface.	**show access-class**

Removing Access Lists from Access Classes

To remove an access list from an interface, remove it from the access class on that interface. To do so, perform the following task within interface <c/s/i> mode:

Task	Command
Remove an access list from an interface.	**no access-class** <access list number> {**in** | **out**} **priority** <number>

Packet Filtering Considerations and Example

This section reviews the considerations you should keep in mind when creating packet filters, and provides examples using the **access-list** and **access-class** commands.

When configuring packet filtering, consider the information in the next two sections.

Implicit Deny

Access lists contain an implicit deny at the end. This means packets for which no match is found are rejected. When more than one access list is applied to an interface, non-matching packets are compared to the access-list with the next highest priority. If a match is still not found, the packet is matched against the next access list. If, after applying the packet to the final access list on an interface, a match is not found the packet is dropped.

For example, the following access list will cause all IP packets to be rejected:

```
cli:172.16.19.10:interface:pos:csi(1/3/1)# access-list
2 deny 1 ip 172.16.19.20 0.0.0.0 any
```

Because that is the only rule defined in the list, the composition of access list 2 is currently:

```
deny 172.16.19.20 any
deny any any
```

All packets are rejected because any 172.16.19.20 source destination is denied, as well as any packets not matching the 172.16.19.20 due to the implicit deny. To solve this, place a permit any statement in the list as follows:

```
cli:172.16.19.10:interface:pos:csi(1/3/1)# access-list
2 permit 2 ip any any
```

Access list 2 is now comprised of the following rules:

```
deny 172.16.19.20 any
permit any any
deny any any
```

With the added *permit any* rule, only packets from the 172.16.19.20 are rejected, all others pass. This is because once the *permit any* condition is met, no further lines in the access list are read.

Match Sequence

The sequence in which an inbound or outbound packet is matched against the filter criteria of an interface is determined by the following:

- **Rule number within access list** — Lower rule numbers take precedence over higher rule numbers. This means that within an access list, the rule with the lower number is examined first.
- P**riority of access-list within the access class** — When you apply an access-list to an interface, access lists assigned lower priorities take precedence over lists assigned higher priorities. This means that within an access class, the access list with the lower number is examined first.

Sample Access List

The following example configures cable interface 1/1/1 to permit all IP traffic except Telnet (TCP 23):

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# access-list 1 deny 1 tcp any eq 23
any any
cli:172.16.19.10:interface:cable:csi(1/1/1)# access-list 1 permit 2 ip any any
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-list 1

Ac Rule Prot IP Source       Mask Source     Start End   Estab TOS ChTOS
             IP Dest         Mask Dest       Port  Port  Sync  Msk
-- ---- ---- --------------- --------------- ----- ----- ----- --- ------
DE    1  tcp         0.0.0.0 255.255.255.255    23    23 False   0      0
                    0.0.0.0 255.255.255.255     0 65535 False   0
.. .... .... ............... ............... ..... ..... ..... ... ......
PE    2   ip         0.0.0.0 255.255.255.255             False   0      0
                    0.0.0.0 255.255.255.255             False   0
.. .... .... ............... ............... ..... ..... ..... ... ......

row count: 2
cli:172.16.19.10:interface:cable:csi(1/1/1)# access-class 1 in priority 1
cli:172.16.19.10:interface:cable:csi(1/1/1)# show access-class

Access List      Priority Row
List   Direction          Status
Number
------ --------- -------- ------
     1        in        1      1

row count: 1
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

16 NETWORK-LAYER BRIDGING

Network-layer bridging allows a single subnet to span across multiple DOCSIS modules. This chapter provides information and procedures about network-layer bridging on the Cuda 12000 and includes the following sections:

About Network-Layer Bridging

Network-layer bridging allows you to add the same IP address to multiple physical interfaces throughout the system. Of particular value is the ability to propagate the same IP gateway across cable interfaces on multiple DOCSIS (CMTS) modules.

The cable modem, customer premise equipment (CPE), or Multimedia Terminal Adapter (MTA) gateway determines the subnet to which a modem, CPE, or MTA can belong. When the provisioning server receives a DHCP request from a cable modem, CPE device, or MTA, it uses the cable modem, CPE, or MTA gateway as a key to determine from which subnet or subnet pool to assign an address.

Routing logic dictates that each interface in the system must have a unique IP address. Network-layer bridging support allows you to group multiple interfaces residing on multiple modules into a single logical interface, known as a *bridge group.* After you assign an IP address to this bridge group, the address will apply to all interfaces that are members of the bridge group.

Creating Network-Layer Bridges

The key to spanning a single subnet across multiple DOCSIS modules is to configure the same IP gateway on each module. Because the gateway serves as the key that dictates address assignment for cable modems, CPE devices or MTAs, configuring the same IP gateway on each cable interface enables the DHCP server to assign those devices IP addresses from the same subnet or subnet pool.

For example, this means that cable modems attached to a DOCSIS module in slot 1 can belong to the same subnet as the modems attached to a DOCSIS module in slot 8. You can then physically move modems between DOCSIS module without assigning new addresses; the shift of modems between modules becomes plug and play.

To span a subnet across multiple cable interfaces, perform the following steps:

1. Create a bridge group, as described in "Creating Bridge Groups" on page 347.
2. Add the cable interfaces on which you want to install the same gateway to the bridge group, as described in "Adding Interfaces to Bridge Groups" on page 349.
3. Assign the IP address that you want to use as the cable modem, CPE, and MTA gateway to the bridge group, as described in "Assigning IP Addresses To Bridge Groups" on page 351. Note that the address that you assign to the bridge group is automatically added to the routing table.
4. Configure the DHCP relay agent on each cable interface so that the IP address is configured as the cable modem, CPE, and/or MTA gateway.

The system supports network-layer bridging within a single chassis where egress ports within the chassis share an IP address. It also supports network-layer bridging within a cluster where egress ports on modules that reside in different chassis can share an IP address. In this way, the layer 3 bridge can span across a single chassis, or multiple chassis in the same cluster.

If for any reason you would like to assign the same IP address for non-cable interfaces, note that you can also add Ethernet and Gigabit Ethernet interfaces to bridge groups.

- POS interfaces cannot be added to a network-layer bridge.
- You can only assign 1 IP address to a specified bridge group.

A single egress port can belong to a maximum of 16 different NLBGs. An NLBG can contain up to 32 physical interfaces; you can define a maximum of 16 NLBGs on single chassis.

Creating Bridge Groups

You must first create a network-layer bridge group before you can configure it. After you create the bridge group, you then configure it within interface configuration mode.

You can identify bridge groups using either numbers or strings; the text string that you specify is case-sensitive. To create a network-layer bridge, perform the following task within any mode:

Task	Command
Create a bridge group.	**bridge-group** <string>

The following example creates a network-layer bridge called Bridge_1, then uses the **show bridge-group** command to verify its creation:

```
cli:172.16.19.10:root# bridge-group Bridge_1
cli:172.16.19.10:root# show bridge-group

Bridge Group: Bridge_1

cli:172.16.19.10:root#
```

To display network-layer bridges, perform the following tasks within any mode:

Task	Command
1. Display a specified bridge group.	**show bridge-group** <bridge-group>
2. Display all bridge groups within the system.	**show bridge-group**

To remove network-layer bridges, perform the following tasks within any mode:

Task	Command
1. Remove a specified bridge group.	**no bridge-group** <bridge-group>
2. Remove all currently configured bridge groups.	**no bridge-group all**

The follow example removes all bridge groups configured on the system:

```
cli:172.16.19.10:root# show bridge-group

Bridge Group: Bridge_1
Bridge Group: 5

cli:172.16.19.10:root# no bridge-group all
2 Bridge Groups have been deleted
cli:172.16.19.10:root#
```

Adding Interfaces to Bridge Groups

After you create a bridge group, you can then enter configuration mode for the group and assign system interfaces to it. All interfaces that you add to the bridge group become part of the layer 3 bridge.

To add an interface to a specified bridge group, perform the following tasks:

Task	Command
1. Enter interface configuration mode for the bridge group to which you want to add interfaces.	**interface bridge-group** <bridge-group>
2. Specify the interface that you want to add to the group.	**bridge-interface** <*c/s/i*>

The following example adds cable interface 1/1/1, interface 1/11/1, and interface 1/11/8 to Bridge Group 1:

```
cli:172.16.19.10:interface:bridge-group(1)# bridge-interface 1/1/1
cli:172.16.19.10:interface:bridge-group(1)# bridge-interface 1/11/1
cli:172.16.19.10:interface:bridge-group(1)# bridge-interface 1/11/8
cli:172.16.19.10:interface:bridge-group(1)# show bridge-group 1

Bridge Group: 1


Chassis    Slot       Interface
---------- ---------- ----------
         1          1          1
         1         11          1
         1         11          8

row count: 3
cli:172.16.19.10:interface:bridge-group(1)#
```

To remove an interface from a bridge group, use the **no bridge-interface** command, as shown in the following example:

```
cli:172.16.19.10:interface:bridge-group(1)# show bridge-group 1

Bridge Group: 1


Chassis    Slot       Interface
---------- ---------- ----------
         1          1          1
         1         11          1
         1         11          8

row count: 3

cli:172.16.19.10:interface:bridge-group(1)# no bridge-interface 1/11/8
cli:172.16.19.10:interface:bridge-group(1)# show bridge-group

Bridge Group: 1


Chassis    Slot       Interface
---------- ---------- ----------
         1          1          1
         1         11          1

row count: 2
cli:172.16.19.10:interface:bridge-group(1)#
```

Assigning IP Addresses To Bridge Groups

A network-layer bridge is comprised of interfaces that belong to the same bridge group. They share any IP address that you assign to the bridge group. The IP address that you assign to the bridge-group is automatically added to the routing table.

Note that because the routing table is automatically updated upon assigning the IP address to the bridge group, you do not have to specifically install the address on the physical interface.

You assign an IP address to a network-layer bridge just as you would any physical interface — by using the **ip address** command within interface configuration mode. The following example assigns IP address 172.16.19.2 to bridge group 1:

```
cli:172.16.19.10:root# interface bridge-group 1
mode: interface:bridge-group(1)
cli:172.16.19.10:interface:bridge-group(1)# ip address 172.16.19.2 255.255.255.0
cli:172.16.19.10:interface:bridge-group(1):ip-address(172.16.19.2)#
```

17 MANAGING IP MULTICAST

This chapter describes how to manage IP Multicast on the Cuda 12000 and includes the following sections:

About IP Multicast

IP Multicast reduces traffic on a network by delivering a single stream of information to multiple users at one time. The Cuda 12000 supports up to 500 multicast groups per chassis.

IGMP

Internet Group Management Protocol (IGMP) is required by all hosts and routers to receive multicast packets. For a host to receive multicast traffic from a specific multicast group, it must join that multicast group. IP hosts then use IGMP to report multicast group memberships to routers.

You can configure an interface to be an IGMP querier or an IGMP host. An IGMP querier periodically transmits IGMP queries to learn which multicast groups are on a network. An IGMP host receives the queries and replies for each multicast group for which they wish to receive traffic.

One instance of IGMP runs on each interface. Each instance of IGMP retains the information for that interface; it has no knowledge of any multicast groups on other interfaces within the chassis.

IGMP Proxy

IGMP Proxy is a method of informing a multicast router about multicast groups for which hosts, that are not directly connected to the router, want to receive traffic.

Example

For example, in Figure 17-1, each interface except the one connected to the remote multicast router is configured to function as an IGMP querier. The Cuda 12000 sends out IGMP queries on each querier interface. When the attached hosts receive these queries, they rely with IGMP reports for each multicast group for which they want to receive multicast traffic. The reports tell the Cuda 12000 about the multicast groups from which PC4, PC3, PC2 and PC1 want to receive packets. To transmit the multicast traffic to hosts PC4, PC3, PC2, and PC1, the remote multicast router is made aware that a host is requesting traffic for these multicast groups.

If IGMP proxy is enabled for these multicast groups, the Cuda 12000 joins these groups on the proxy interface. When the remote multicast router sends an IGMP query to the Cuda 12000, the Cuda 12000 replies with IGMP reports for all multicast groups joined by the hosts. Since the remote multicast router knows about the additional multicast groups joined by the hosts, it routes multicast traffic for these groups onto the common Ethernet.

When the Cuda 12000 receives this multicast traffic, it forwards the traffic out to all other interfaces that have joined the same multicast group. In the figure, if the remote multicast router receives a packet addressed to 224.17.1.5, it transmits the packet to the common Ethernet. The Cuda 12000 then receives the packet and forwards it to interface 1/2/1 and 1/5/3. PC3 and PC2 receive the multicast packet.

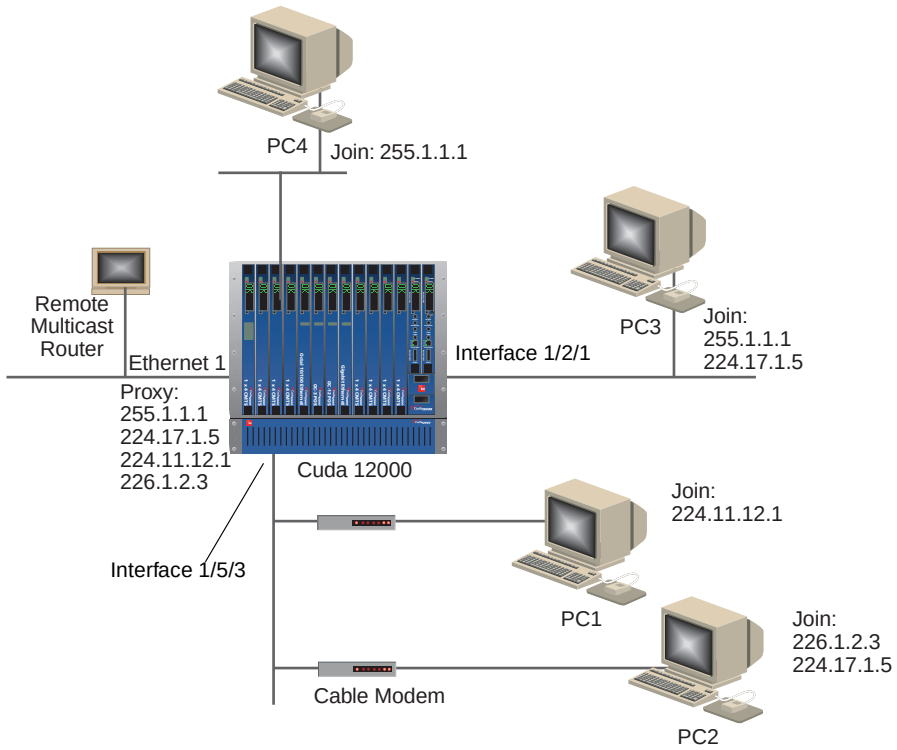


Figure 17-1 Sample Network

In the Cuda 12000, you can configure interfaces to proxy for individual IP multicast addresses or multicast address ranges. This gives you a lot of flexibility. For example, you can proxy for a single multicast group on one interface and a different multicast group on another interface. Alternatively, you could also include these groups in the same range and proxy for them both on the same interface.

Managing IGMP Interfaces

Perform the following tasks to manage IGMP interfaces:

- Join groups
- Configure IGMP interface parameters
- Display groups and parameters
- Delete groups

Joining IGMP Groups

An interface on a Cuda 12000 can join a multicast group in two ways:

- Manually, through the **ip igmp join-group** command
- Automatically, through the reception of IGMP reports from the network

There are no predefined multicast groups. You can configure up to 500 groups per chassis and up to 50 groups per DOCSIS module.

Use the following procedure to join a multicast group on an IP interface manually:

Task	Command
1. Enter configuration mode for the selected interface.	**interface** <c/s/i>
2. Join the group on the interface.	**ip igmp join-group** <group-address>

Configuring IGMP Interface Parameters

You can configure the following IGMP interface parameters:

- **Query Interval** — Specifies the frequency, in seconds, that the Cuda 12000 transmits IGMP host query packets on this particular interface. The default is 125 seconds with a range of 10 to 65535 seconds.
- **Query Max Response Time** — Specifies the maximum number of seconds that the Cuda 12000 waits for a response to an IGMP Query message before deleting the group. The default is 10 seconds with a range of 1 to 25 seconds.
- **Version** — Version of IGMP running on this particular interface. For IGMP to function properly, all routers on a network must be configured to run the same version of IGMP. The default is 2 and the possible values are:
  - 2 – Version 2. If the Cuda 12000 encounters another host or router on the network using Version 1, the Cuda 12000 reverts back to using Version 1.
  - 1 – Version 1.
  - v2_only – Version 2 only. If the Cuda 12000 encounters another host or router on the network using Version 1, the Cuda 12000 continues to run Version 2.
- **Robustness** — Allows you to compensate for the expected packet loss on a subnet. Increase the value if you expect the loss to be high. The default is 2 and the range is 1 to 255.
- **Router** — Enables the interface to function as an IGMP Querier (router). For the DOCSIS module, the default is router. If multiple routers attempt to become the IGMP Querier, the one with the lowest IP address becomes the querier. When you change an interface from a querier to a host, any multicast groups that are learned are removed. When you change the interface from a host to a querier, or from a querier to a host, any multicast groups joined locally or by an application running on the Cuda 12000 remain. Note that the DOCSIS module can only be configured as an IGMP querier (router); all other interfaces default to IGMP hosts.
- **Last Query Interval** — Specifies the number of seconds between queries to find other hosts on the network that want to receive traffic from the multicast group. You can tune this parameter to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group. The default is 1 and the range is 1 to 25 seconds.

Use the following procedure to configure IGMP interface parameters:

Task	Command
1. Enter configuration mode for the selected interface.	**interface** <c/s/i>
2. Set the IGMP interface parameter.	**ip igmp** {**query-interval** <seconds> | **query-max-response time** <seconds> | **version** {**2** | **1** | **v2_only**} | **robustness** <value> | **router** | **last-query-interval** <seconds>}

Displaying IGMP Groups and Interface Parameters

You can display IGMP groups and interface parameters from either root mode or configuration mode.

Displaying Groups and Parameters from Root Mode

Use this procedure to display IGMP groups and interface parameters from root mode:

Task	Command
1. Enter root mode.	**root.**
2. Display groups or interface parameters.	**show ip igmp** {**groups** {<group-address> | <c/s/i>} | **interface** <c/s/i>}

Note that:

- When you issue the **show ip igmp groups** command without specifying a group address or an interface, the command displays all groups on all interfaces.
- When you issue the **show ip igmp interface** command without specifying an interface, the command displays details on all interfaces.

Table 17-1 describes interface details that the **show ip igmp interface** command displays. Table 17-2 displays details that the **show ip igmp groups** command displays.

Table 17-1 Interface Details

Parameter	Description
Multicast forwarding	Indicates whether multicast forwarding is enabled or disabled on the interface.
IP Address	The lowest IP address configured on the specified interface. An IP address of 0.0.0.0 means that the interface functions as an IGMP host. An IP address other than 0.0.0.0 means that this interface is the IGMP Querier on the interface's network.
Interface is	Indicates how IGMP is functioning on this interface. The options are:
IGMP Host	Receives IGMP queries and replies for each multicast group for which it wishes to receive traffic.
IGMP Querier	Periodically transmits IGMP queries to finds multicast groups on a network.
Non-querier	If the current IGMP querier interface stops functioning, the non-querier interface becomes the querier (that is, it acts as a backup querier).
Interface	The interface identifier (for example, 1/1/1).
Querier	IP address of the IGMP querier on the IP subnet to which this interface is attached.
Up Time	Time since the IP address of the IGMP querier changed.
Version	Version of IGMP running on this particular interface. For IGMP to function properly, all routers on a network must be configured to run the same version of IGMP. The default is 2 and the possible values are:
2	Version 2. If the Cuda 12000 encounters another host or router on the network using Version 1, the Cuda 12000 reverts back to using Version 1.
1	Version 1
V2_ONLY	Version 2 only. If the Cuda 12000 encounters another host or router on the network using Version 1, the Cuda 12000 continues to run Version 2.
Query Interval	Frequency, in seconds, that the IGMP host query packets are transmitted on this particular interface. The default is 125 seconds with a range of 10 to 65535 seconds.

Table 17-1 Interface Details (continued)

Parameter	Description
Max Resp Time	Maximum number of seconds to wait for a response to an IGMP Query message before the group is deleted. The default is 10 seconds with a range of 1 to 25.
Robustness	Allows you to compensate for the expected packet loss on a subnet. If the loss is expected to be high, increase the value. The default is 2 and the range is 1 to 255.
Joins	Number of multicast groups joined on this interface since it was enabled. This parameter reflects the amount of IGMP activity.
Wrong Queries	Read-only. Number of queries received indicating that the IGMP version does not match the **Version** value configured on this interface. IGMP requires all routers on a network to be configured to operate with the same version of IGMP. If any queries indicate the wrong version, this indicates a configuration error.
Groups	Number of current IGMP groups joined on this interface.
Last Member (Query) Interval	A query is sent to determine if other hosts on the network wish to receive traffic from the multicast group. The **Last Query Interval** is the time between queries. You can tune this parameter to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group. The parameter is ignored for Version 1 of IGMP. The default is 1 and the range is 1 to 25 seconds.
Version 1 Querier Timer	Remaining time until the Cuda 12000 determines that no IGMPv1 routers are present on the interface. When the value is greater than 0, the host replies to all queries with IGMPv1 membership reports.

Table 17-2 Group Details

Parameter	Description
Group Address	The IP address of the IGMP group.
Up Time	Time elapsed in hours, minutes, and seconds since the creation of the entry.
Expires	Minimum amount of time remaining before this entry is aged out. If the value is zero, the entry does not time out.
Last Reporter	Source IP address for the last membership report received for this group IP address. If no report is received, the value is 0.0.0.0.
Status	Status of the entry including:
Learned	Group is learned by receiving an IGMP report over the network
Self	Group is locally joined (for example, joined manually from the CLI) or joined from an application on the Cuda 12000 such as OSPF or RIP.
Proxy	Group is being proxied on this interface.

Displaying Groups and Parameters from Interface Mode

To display IGMP groups and interface parameters from interface mode:

Task	Command
1. Enter configuration mode for the interface.	**interface** <c/s/i>
2. Display groups or interface parameters.	**show ip igmp groups** {<group-address> | <c/s/i>} | **interface** <c/s/i>}

Note that

- Issue the **show ip igmp groups** command without specifying a group address or an interface to display all groups on the current interface.
- Issue the **show ip igmp interface** command without specifying an interface to display details on the current interface.

See Table 17-1 for descriptions of interface details. See Table 17-2 for descriptions of group details.

Deleting IGMP Groups

You can delete IGMP groups that you join on an interface using the **ip igmp join-group** command. You can delete these groups in two ways:

- Using the **no ip igmp join-group** command
- Using the **clear ip igmp group** command

To delete groups with the **no ip igmp join-group** command:

Task	Command
1. Enter interface mode.	**root**
2. Display groups.	**show ip igmp groups**
3. Specify the multicast address of the group you want to delete.	**no ip igmp groups** <group-address>

To delete multicast group entries with the **clear ip igmp group** command:

Task	Command
1. Enter root mode or interface mode.	**enable root** or **interface** <c/s/i>
2. Display multicast groups.	**show ip igmp groups**
3. Specify arguments as follows: ■ To flush the entire cache on the Cuda 12000, issue the command in root mode without specifying a group address. ■ To remove a single multicast group from all interfaces on the Cuda 12000, issue the command in root mode and specify the address of the group. ■ To remove all multicast groups from an interface, issue the command in interface mode without specifying a group address. ■ To remove a single multicast group from an interface, issue the command in interface mode and specify the address of the group.	**clear ip igmp group** [<group-address>]

Managing IGMP Proxies

You can configure an interface to proxy for a single multicast group or a range of multicast groups. You can also display and delete IGMP proxies.

Configuring Proxies

Before configuring an interface to proxy for multicast groups, note that:

- You must assign an IP address to that interface.
- You cannot configure an interface to proxy for a multicast group within the multicast range 224.0.0.0 to 224.0.0.255.
- If the interface is configured to proxy for multiple multicast groups, the most specific match acts as the proxy for that interface. However, if the interface proxies for the same multicast address ranges, the metric value determines which address range is used.

Use this procedure to configure interfaces to proxy for multicast groups:

Task	Command
1. Enter root mode.	**root**
2. Specify the multicast address or address range, mask, metric, and interface that will act as the proxy.	**ip igmp proxy** <group-address> <group-mask> **metric** <number> <c/s/i>

Examples

You can configure an interface to proxy for a single multicast group or a range of multicast groups. An example of each instance is shown below:

Example 1 — This example shows a range of multicast groups for which the interface proxies:

- **Group Address** — 225.1.0.0
- **Mask** — 255.255.0.0

The interface proxies for an address range from 225.1.0.0 to 255.1.255.255

Example 2 — This example shows a single multicast group for which the interface proxies:

- **Group Address** — 226.1.1.1
- **Mask** — 255.255.255.255

The interface proxies for the multicast group 226.1.1.1.

Displaying Proxies

Use this procedure to display proxies for multicast groups:

Task	Command
1. Enter root mode.	**root**
2. Display proxies.	**show ip igmp proxy**

The **show ip igmp proxy** command displays the following information about each proxy:

- **Group Address** — IP Multicast group address or address range for which the interface acts as a proxy.
- **Mask** — Mask associated with the group address.
- **Interface** — Interface that acts as the proxy.
- **Metric** — Metric value from 1 to 255 that assigns a priority to the proxy. One (1) is the highest priority; 255 is the lowest priority.
- **Status** — Active or Backup. Active status means that the proxy is currently in use. Backup status means that the proxy is currently not in use.

Deleting Proxies

Use this procedure to delete proxies for multicast groups:

Task	Command
1. Enter root mode.	**root**
2. Display proxies.	**show ip igmp proxy**
3. Specify the multicast address or address range, mask, and metric for the proxy that you want to delete.	**no ip igmp proxy** <group-address> <group-mask> **metric** <number>

Displaying Multicast Routes

Use this procedure to display multicast routes:

Task	Command
1. Enter root mode or interface mode.	**enable root** or **interface** <c/s/i>
2. Display summary information on multicast routes, details on multicast routes, or details on a specific multicast route.	**show ip igmp mroute** {<group-address> **| summary**}

When issued with the **summary** argument, the command provides this information:

- **MRoute Group** — IP Multicast group address that contains the multicast routing information.
- **Up Time** — Time in hours, minutes, and seconds since the multicast routing information was learned.

When issued with no argument or the <group-address> argument, the command provides this information in addition to the summary information:

- **Outgoing Interface** — Interface on which the multicast route is learned or joined.
- **Outgoing Interface Up Time** — Time in hours, minutes, and seconds since the multicast routing information was learned.

IV Cable Modem Termination Systems

18 CONFIGURING CABLE MODEM TERMINATION SYSTEMS

Cable Modem Termination Systems (CMTS) consists of DOCSIS and EuroDOCSIS modules within the Cuda 12000. Configuring a CMTS consists of the following functions:

- Configuring the MAC Interface (page 370)
- Configuring the Downstream Channel (page 379)
- Configuring Upstream Channels (page 390)
- Configuring Admission Control (page 408)
- Configuring Frequency Hopping (page 411)
- Defining Modulation Profiles (page 418)
- Configuring CMTS Privacy Parameters (page 428)
- Configuring Flap Control (page 428)

Before you can effectively perform these tasks, you need an understanding of CMTS upstream frequency reuses.

CMTS Upstream Frequency Reuse

The Cuda 12000 supports the configuration of upstream channels with the same center frequencies, if the channels are on separate non-combined physical plants. Referred to as *upstream frequency reuse*, this allows an operator to set aside less of the valuable upstream spectrum for CMTS use. Note, however, that for proper operation upstream channels must also be configured to have the same channel widths and minislot sizes.

Configuring the MAC Interface

Media Access Control (MAC) is a logical interface implemented within hardware and software. MAC contains one downstream and four upstream channels. Frequencies are assigned for each of the downstream and upstream channels.

MAC interface parameters are associated with DOCSIS/EuroDOCSIS module timing and control features that exist at network layer 2 to manage upstream and downstream traffic.

Displaying MAC Interface Parameters and Statistics

You can display CMTS MAC interface configuration and statistics for a specific cable interface by performing the following tasks:

Task	Command
1. Display MAC parameters and statistics for the specified cable interface from within root or cable interface configuration mode.	**show interface cable** <c/s/i> **mac**
2. Display MAC parameters and statistics for the current cable interface from the cable interface configuration mode.	**show mac**

Example

The following example displays the current MAC configuration and related statistics for cable interface 1/1/1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show mac
Insertion Interval                          10  (centisec)
Invited Ranging Attempts                    16
Sync Interval                                5  (millisec)
UCD Interval                              2000  (millisec)

Hardware Map Timer                        2000  (microsec)
Periodic Ranging Timer                      15  (secs)
Plant Delay                               1600

PLL State                                normal
PLL Value                                     0
Stats:
Admin Status                                 up
Operational Status                           up
In Octets                                838461
In Unicast Packets                        15502
In Multicacast Packets                        0
In Broadcast Packets                          1
In Errors                                     0
In Discards                                   0
Out Octets                           606211931
Out Unicast Packets                      13871
Out Multicacast Packets                  13619640
Out Broadcast Packets                      283
Out Errors                                    0
Out Discards                                  0

Invalid Range Requests                        0
Ranging Aborts                                1
Invalid Registration Reque                    0
Failed Registration Reques                    0
Invalid Data Requests                         0
T5 Timeouts                                   0

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Understanding MAC Interface Statistics

MAC interface statistics are displayed as part of the **show interface cable** <c/s/i> **mac** display, as shown in the previous section. Table 18-1 provides a brief description for each MAC statistic.

Table 18-1 CMTS MAC Interface Statistics

Statistic	Description
In Octets	Aggregate number of bytes received on all upstream channels.
In Unicast Packets	Aggregate number of unicast packets received on all upstream channels.
In Multicast Packets	Aggregate number of multicast packets received on all upstream channels.
In Broadcast Packets	Aggregate number of broadcast packets received on all upstream channels.
In Errors	Aggregate number of error packets received on all upstream channels.
In Discards	Aggregate number of discard packets received on all upstream channels.
Out Octets	Number of bytes transmitted from the downstream channel.
Out Unicast Packets	Displays the number of unicast packets transmitted from the downstream channel.
Out Multicast Packets	Number of multicast packets transmitted from the downstream channel.
Out Broadcast Packets	Number of broadcast packets transmitted from the downstream channel.
Out Errors	Aggregate number of error packets transmitted from the downstream channel.
Out Discards	Aggregate number of packets discarded on the downstream channel.

Statistic	Description
Invalid Range Requests	Aggregate number of invalid ranging requests received on the MAC interface.
Ranging Aborts	Number of abort range responses that were sent by the CMTS.
Invalid Registration Requests	Aggregate number of invalid registration requests received on the MAC interface.
Failed Regestration Requests	Aggregate number of failed registration requests from modems.
Invalid Data Requests	Aggregate number of invalid data requests received on the MAC interface.
T5 Timeouts	Number of timeouts that occurred while waiting for upstream channel change responses.

Configuring MAC Interface Parameters

MAC interface parameters are described in the following sections.

Shared Secret

The Shared Secret parameter is the authentication string shared between the CMTS and the provisioning server. The shared secret is used by the CMTS to validate that a cable modem was provisioned by an authorized server. If this parameter is left blank, the CMTS does not validate a modem's provisioning. Note that the same value must be configured on the provisioning server. Configuring no shared secret turns off authentication of modem provisioning.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the shared-secret for the current cable interface.	**shared-secret** [**ascii**] <secret-string>
3. Display the shared secret.	**show interface cable** <c/s/i> **shared-secret** or *Within cable interface mode:* **show shared-secret**

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# shared-secret
123456
cli:172.16.19.10:interface:cable:csi(1/1/1)# show shared-secret
Shared Key
     ASCII: "4V"
     HEX:   "12:34:56"
```

Sync Interval (millisec)

The Sync Interval parameters sets the time interval between the CMTS transmission of SYNC messages. By default, the SYNC message is sent by the MAC hardware every 5 milliseconds. Acceptable values are 1 to 200 milliseconds.

You set the sync interval by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the CMTS ranging sync interval.	**sync-interval** <value>

UCD-Interval

The UCD (Upstream Channel Descriptor) Interval sets the time interval between CMTS transmission of Upstream Channel Descriptor for each active upstream channel. By default, the UCD is sent every 2000 milliseconds. Acceptable values are 1 to 2000 milliseconds.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the CMTS ranging UCD interval.	**ucd-interval** <value>

Insertion-interval

The Insertion-interval parameter specifies the interval between CMTS transmission of Initial Maintenance (IM) intervals. This limits the amount of time during which cable modems can request an upstream frequency from the CMTS and join the network for the first time. By default, the automatic setting is configured at 10 centiseconds. Acceptable values are 5 to 200 centiseconds.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the CMTS ranging insertion interval.	**insertion-interval** <value>

Ranging-attempts

The Ranging-attempts parameter specifies the maximum number of attempts for the CMTS to attempt ranging a modem as of tolerance or not responding. A value of zero means the system should attempt to range forever. By default, attempts are sent every 16 attempts per ranging period as defined by the Periodic Ranging Timer. Acceptable values are 0 – 1024.

You set the CMTS Ranging-attempts parameter by performing the following tasks:

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the CMTS ranging attempts.	**ranging-attempts** <value>

Map-timer

The Map-timer parameter sets the time interval between the CMTS transmission of MAP messages for each active upstream channel. Acceptable values are 1000 – 10000 microseconds. The default is 10 milliseconds. Setting this value at less than 6 milliseconds causes performance problems.

Perform the following task to set the Map-timer parameter for the current cable interface.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure Map Timer.	**map-timer** <value>

Periodic-Ranging-Interval

The Periodic-Ranging-Interval parameter defines the period during which the CMTS will offer a ranging opportunity to each cable modem. By default, Periodic Ranging is sent every 15 seconds. Acceptable values are 5 to 30 seconds.

You set the Periodic-ranging-interval for the current cable interface by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure Periodic Ranging Timer.	**periodic-ranging-interval** <value>

Plant-delay

The Plant-delay parameter specifies the maximum round-trip propagation delay in the cable plant. This value is used to adjust the map lead time. It is recommended that a low value be used to reduce cable modem access delay.

- For a cable plant of 25 miles, the recommended value is 400.
- For a cable plant of 100 miles, the recommended value is 1600.

You set the plant delay for the current cable interface by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure plant delay.	**plant-delay** <value>

PLL State

Phase-locked loops (PLL) are circuits that hunt and synchronize to an external signal. The PLL State value should be zero; this indicates normal operation. *A PLL State value of non-zero, indicates a malfunction of the CMTS module.*

The PLL state is displayed as part of the **show interface cable** <c/s/i> **mac** display. You can also verify the PLL State by performing the following task.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Verify PLL state of the current cable interface.	**show interface cable** <c/s/i> **pll-state**

Configuring the Downstream Channel

The Downstream Channel sends data from the headend site to subscriber cable modems. Configuring the downstream channel involves setting parameters to maximize the performance of the data transfer. Downstream channel parameters are based on the modulation type for a downstream channel on the CMTS.

The downstream center frequency range values and the downstream interleave depth values are different for DOCSIS and EuroDOCSIS. Refer to the descriptions below for detailed configuration information.

Displaying Downstream Configuration and Statistics

You display CMTS downstream channel configuration and statistics for a specific cable interface by performing one of the following tasks:

Task	Command
1. Enter this command within root mode or cable interface configuration mode to display downstream channel parameters and statistics.	**show interface cable** <c/s/i> **downstream**
2. Enter this command within cable interface configuration mode to display downstream parameters and statistics for the current interface.	**show downstream**

Example

The following example displays the current downstream configuration and related statistics for cable interface 1/1/1:

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show downstream
C/S/P                          1 / 1 / 2 / 2
Frequency                              507.0  (MHz)
Interleave                 taps32Increment4
Modulation                           qam256
ChannelWidth                              6  (MHz)
ChannelPower                            550  (1/10 dBmV)
AnnexType                           Annex B

Symbol Rate                          5360537  (baud)

Admin Status                              up
Operational Status                        up

Out Octets                         629121492
Out Unicast Packets                    14416
Out Multicacast Packets             14134320
Out Broadcast Packets                    288
Out Errors                                 0
Out Discards                               0

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The Annex Type for EuroDOCSIS modules is Annex A.

Understanding Downstream Channel Statistics

Downstream statistics are displayed as part of the show interface cable <c/s/i> downstream display as shown in the previous section. Table 18-2 provides a brief description for each MAC statistic.

Table 18-2 Downstream Channel Statistics

Statistic	Description
Symbol Rate	Specifies the MAC symbol rate in symbols per second: ■ **qam64** – 5,056,941 symbols per second. ■ **qam256** – 5,360,537 symbols per second.
Out Octets	The number of bytes transmitted on the interface.
Unicast Packets	The number of unicast packets transmitted on the downstream channel.
Multicast Packets	The number of multicast packets transmitted on the downstream channel.
Broadcast Packets	The number of broadcast packets transmitted on the downstream channel
Discard Packets	Aggregate number of discard packets transmitted on the downstream channel.
Error Packets	Aggregate number of error packets transmitted on the downstream channel.

Configuring Downstream Parameters

Downstream channel configuration is described in the following sections:

Annex Type

The downstream channel Annex Type parameter supports MPEG framing format for DOCSIS and EuroDOCSIS modules. The Cuda 12000 automatically detects MPEG framing format, as follows:

- **Annex A** — Indicates an MPEG framing format for a EuroDOCSIS module.
- **Annex B** — Indicates an MPEG framing format for a DOCSIS module.

Downstream Shutdown

The Downstream Shutdown parameter sets the state of the downstream channel to *up* or *down*. **Up** indicates that the channel is active; **down** indicates that the channel is inactive. By default, the channel should be down, unless a prior configuration exists with the channel in the active state.

Perform the following tasks to set the downstream channel status.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the downstream status to up.	**downstream no shutdown**
3. Set the downstream status to down.	**downstream shutdown**

Downstream Frequency

The Downstream Frequency (Hz) parameter sets the downstream signal for the RF carrier. By default, Center Frequency is set at 507 MHz. *DOCSIS* acceptable values are 93 MHz to 855 MHz; *EuroDOCSIS* acceptable values are 91.0 MHz to 858 MHz.

You set the downstream center frequency by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the downstream center frequency.	**downstream frequency** <freq-number>

Channel Width

The Channel Width (MHz) parameter is fixed based on DOCSIS standards. By default, the channel is set at 6 MHz, which is an acceptable value defined by the NTSC channel plan. EuroDOCSIS modules operate over 8 MHz channels. Channel width is shown as part of the **show interface cable** <c/s/i> **downstream** display.

Downstream transmit-power

The Downstream transmit-power (TenthdBmV) parameter sets the nominal output transmit power level. By default, Channel Power is set at 550 TenthdBmV. Acceptable values: 0–650.

You set the downstream output transmit power level by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the output transmit power level.	**downstream transmit-power** <0..650>

Downstream Modulation

The Downstream modulation parameter sets the modulation rate for a downstream channel. The CMTS supports the following two modulation types.

You set the downstream modulation type by performing the following tasks.

Task	Command
1. Enter interface configuration mode.	**interface cable** <c/s/i>
2. Configure modulation used on the downstream channel. Valid values: ■ **qam64**: Sets the interface speed at 30 Mbps *or* ■ **qam256**: Sets the interface speed at 40 Mbps	**downstream modulation** {**qam64** | **qam256**}

Downstream interleave-depth

The Downstream interleave-depth parameter sets the FEC Interleaving for the downstream channel. By default, the Downstream interleave-depth for DOCSIS is set at taps32Increment4. A higher value improves protection from noise bursts; however, it may slow down the downstream data transfer rate.

The Downstream interleave-depth for EuroDOCSIS *must* be set at taps12Increment7.

NOTE: *The first time you install a EuroDOCSIS module you must set the interleave depth to taps12Increment7, in order for the cable modems to register with the downstream channel.*

You set the downstream interleave depth by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the DOCSIS downstream interleave depth. Valid interleave depth values: ■ taps8Increment16 ■ taps16Increment 8 ■ taps32Increment4 ■ taps64Increment2 ■ taps128Increment1	**downstream interleave-depth** {**8** | **16** | **32** | **64** | **128**}
3. Configure the EuroDOCSIS downstream interleave depth. The valid interleave depth value is: taps12Increment7	**downstream interleave-depth 12**

Example Procedure of Downstream Configuration

The following procedure steps you through the process of configuring a DOCSIS downstream channel. An example of a DOCSIS configuration follows:

Task	Command
1. View a list of CMTS interfaces that you have installed on your chassis. You can do so by using a combination BAS/UNIX command.	**show topology | include docs**
2. Enter configuration mode for the CMTS interface that you want to configure.	**interface cable** <c/s/i>
3. Display the current downstream configuration for this CMTS card.	**show interface cable** <c/s/i> **downstream**
4. Set the downstream channel status.	■ **downstream no shutdown** (Sets the downstream channel status to up.) ■ **downstream shutdown** (Sets the downstream channel status to down.)
5. Set the downstream center frequency. Acceptable DOCSIS range: 93.0 MHz to 855 MHz.	**downstream frequency** <freq number>
6. Set the downstream channel interleave depth	**downstream interleave-depth** {**8 | 16 | 32 | 64 | 128**}

Task	Command
7. Set the downstream channel modulation type.	**downstream modulation** {**qam64 | qam256**}
8. Set the downstream channel power. Acceptable values: 0 – 650.	**downstream transmit-power** <value>
9. Verify the downstream configuration for this CMTS card.	**show interface cable** <c/s/i> **downstream**

Example

The following example shows a downstream channel configuration.

```
cli:172.16.19.10:root# show topology | include docs
1 / 1 / 1        Egress     docsCableMaclayer    Active
1 / 6 / 1        Egress     docsCableMaclayer    Active

cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream no shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream frequency 700
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream interleave-depth 32
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream modulation qam256
cli:172.16.19.10:interface:cable:csi(1/1/1)# downstream transmit-power 500
cli:172.16.19.10:interface:cable:csi(1/1/1)# show downstream
C/S/P                           1 / 1 / 2 / 2
Frequency                               700.0  (MHz)
Interleave                  taps32Increment4
Modulation                            qam256
ChannelWidth                               6  (MHz)
ChannelPower                             500  (1/10 dBmV)
AnnexType                            Annex B

Symbol Rate                            5360537  (baud)

Admin Status                                up
Operational Status                          up

Out Octets                           678884867
Out Unicast Packets                     15578
Out Multicacast Packets              15252214
Out Broadcast Packets                     312
Out Errors                                  0
Out Discards                                0

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Configuring Upstream Channels

Upstream channels are used to transfer data from subscriber cable modems to the headend site. Data transfer is accomplished in bursts. The Cuda 12000 supports four upstream channels per DOCSIS/EuroDOCSIS module.

Displaying Upstream Configuration and Statistics

You can display configuration and statistics for DOCSIS/EuroDOCSIS upstream channels, including signal quality information. To do so, perform the following tasks:

Task	Command
1. Enter this command from root mode or cable interface mode to display parameters and statistics for all upstream channels on the specified cable interface.	**show interface cable** <c/s/i> **upstream**
2. Enter this command from within cable interface mode to display parameters and statistics for all upstream channels on the current cable interface.	**show upstream**
3. Enter this command from within root mode or cable interface mode to display parameters and statistics for a specific upstream channel on the specified cable interface. Channel IDs range from 1 to 4 (for 1x4 modules) or from 1 to 6 (for 1x6 modules).	**show interface cable** <c/s/i> **upstream** <port number>

Task	Command
4. Enter this command within cable interface mode to display parameters and statistics for a specific upstream channel on the current cable interface.	**show upstream** <port number>

Example

```
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1
Upstream Channel Id            1 / 1 / 3 / 2
Center Frequency                        20.0  (MHz)
Channel Width                            3.2  (MHz)
Slot Size                                  2  (uSec)
Receive Power                              0  (1/10 dBmV)
Modulation Profile                         1
Tx Timing Offset                        1496
Tx Backoff Start                           5
Tx Backoff End                            10
Ranging Backoff Start                      2
Ranging Backoff End                        3

Admin Status                              up
Operational Status                        up

Stats:
In Octets                             346263
In Unicast Packets                      7281
In Multicacast Packets                     0
In Broadcast Packets                       0
In Errors                                  0
In Discards                                0

Signal Quality:
packets error-free                      8267
corrected                                  0
uncorrected                                0
Signal Noise                             329  (dB)
Microreflections                          11
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Configuring Upstream Channel Parameters

Configuring upstream channel parameters is described in the following sections.

Upstream shutdown

The Upstream shutdown parameter sets the state of the upstream channel. Choose **Up** to set the channel active, or choose **Down** to set the channel inactive.

You set the upstream channel status by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the upstream channel status to up. or Set the upstream channel status to down.	**upstream** <port number> **no shutdown** **upstream** <port number> **shutdown**

Frequency

The Frequency (MHz) parameter sets the upstream signal frequency for the RF carrier. You may choose an acceptable DOCSIS range from 5.0 - 42.0 MHz; or an acceptable EuroDOCSIS range from 5.0 - 65.0 MHz.

You set the upstream channel status by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the upstream channel frequency.	**upstream** <port number> **frequency** <value>

Channel Width (kHz)

The Channel Width parameter sets the upstream channel width in kilohertz (kHz). By default, the Channel Width is set at 3200 kHz (2560 kilosymbols - ksyms). Acceptable values include:

- 200K (160 ksyms per second)
- 400K (320 ksyms per second)
- 800K (640 ksyms per second)
- 1600K (1280 ksyms per second)
- 3200K (2560 ksyms per second)

When modifying the channel width, consider the following:

- The symbol rate is recomputed as follows:

  symbol rate = channel width/1.25
- A higher symbol rate is more susceptible to RF noise and interference.
- If you use a symbol rate or modulation format beyond the capabilities of your HFC network, packet loss or loss of cable modem connectivity may occur.

Perform the following task within interface:cable:csi <c/s/i> mode to set the upstream channel width.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the upstream channel width.	**upstream** <port number> **channel-width** {**200 | 400 | 800 | 1600 | 3200**}

Slot Size

The Slot Size parameter is the number of 6.25 microsecond ticks in each upstream minislot. This depends on one selected channel width, which is automatically set when the user selects an acceptable channel width. By default, the Slot Size is set at 2.

Following are recommended minislot values for different channel widths:

- 2 (3200 kHz)
- 4 (1600 kHz)
- 8 (800 kHz)
- 16 (400 kHz)
- 32 (200 kHz)

WARNING: The slot size affects the performance of the CMTS. It is recommended that configuration is done by an expert-level administrator.

You set the upstream slot size by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the upstream channel mini-slot size.	**upstream** <port number> **minislot-size** {**2** | **4** | **8** | **16** | **32** | **64** | **128**}

Receive Power

The Receive Power parameter sets the level for the upstream interface in TenthdBmV. By default, the Receive Power is set at 0, which is the optimal setting for the upstream power level. Acceptable values are -160 to 260 TenthdBmV. The Receive Power is dependent on the selected channel-width.

You set the upstream receive power by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Set the upstream receive power level.	**upstream** <port number> **power-level** <-160..260>

Modulation Profile

The Profile Index number that identifies the properties of the Upstream Channel ID.

You assign a modulation profile to an upstream channel by performing the following tasks.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Specify the modulation profile that you want the channel to utilize.	**upstream** <port number> **modulation-profile** <**profile number**>

Data-Backoff

The data-backoff parameter sets a fixed start value for initial data backoff on the upstream channels. By default, the TX data backoff Start parameter is set to 5. Acceptable values are 0 to 15.

You set the data backoff range on the upstream channel by performing the following task.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the transmit backoff range.	**upstream** <port number> **data-backoff** <start> <end>

Range-Backoff

The Range Backoff Start parameter sets the fixed start value for range backoff on the upstream channels. By default the start value is set to 2 and the end value is set to 10. Acceptable values are 0 to 15.

You set ranging backoff values on the upstream channel by performing the following task.

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Configure the ranging backoff for an upstream channel.	**upstream** <port number> **range-backoff** <start> <end>

Example Procedure of Upstream Configuration

The following procedure steps you through the process of configuring upstream channel 1. An upstream configuration example for a DOCSIS cable interface follows. The process is similar for the remaining upstream channels.

Task	Command
1. View a list of DOCSIS interfaces that you have installed on your chassis.	**show topology | include docs**
2. Enter configuration mode for the DOCSIS interface that you want to configure.	**interface** <c/s/i>
3. Display the current configuration for the upstream channel that you want to configure.	**show interface cable** <c/s/i> **upstream** [<port number>]
4. Set the channel status for the specified upstream channel.	■ **upstream** <port number> **no shutdown** (Sets the upstream channel status to up.) ■ **upstream** <port number> **shutdown** (Sets the upstream channel status to down.)
5. Set the data backoff for the upstream channel. Valid values:0 –15.	**upstream** <port number> **data-backoff** <start-number> <end-number>
6. Set the range backoff for the selected upstream channel. Valid values: 0 – 15.	**upstream** <port number> **range-backoff** <start-number> <end-number>

Task	Command
7. Set the upstream channel frequency. Valid range: 5.0 – 42.0 MHz.	**upstream** <port number> **frequency** <frequency>
8. Set the mini-slot size for the downstream channel.	**upstream** <port number> **minislot-size** {**2 | 4 | 8 | 16 | 32 | 64 | 128**}
9. Set the receive power level. Acceptable range: -160 – 260 TenthdBmV.	**upstream** <port number> **power-level** <value>
10. Set the upstream channel width.	**upstream** <port number> **channel-width** {**200 | 400 | 800 | 1600 | 3200**}
11. Specify the modulation profile. Acceptable profile numbers:1 – 2.	**upstream** <port number> **modulation-profile** <profile-number>
12. Verify the upstream configuration for the selected channel.	**show interface cable** <c/s/i> **upstream** [<port number>]

Example

The following example configures upstream channel 1 on the CMTS interface installed in slot 1.

```
cli:172.16.19.10:root# show topology | include docs
1 / 1 / 1        Egress     docsCableMaclayer    Active
1 / 6 / 1        Egress     docsCableMaclayer    Active
cli:172.16.19.10:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 no shutdown
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 data-backoff 3 10
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 range-backoff 1 2
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 frequency 320
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 frequency 32
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 minislot-size 8
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 power-level 250
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 channel-width 200
cli:172.16.19.10:interface:cable:csi(1/1/1)# upstream 1 modulation-profile 2
cli:172.16.19.10:interface:cable:csi(1/1/1)# show upstream 1
Upstream Channel Id            1 / 1 / 3 / 2
Center Frequency                        32.0  (MHz)
Channel Width                            0.2  (MHz)
Slot Size                                  8  (uSec)
Receive Power                            250  (1/10 dBmV)
Modulation Profile                         2
Tx Timing Offset                        2586
Tx Backoff Start                           3
Tx Backoff End                            10
Ranging Backoff Start                      1
Ranging Backoff End                        2

Admin Status                              up
Operational Status                        up

Stats:
In Octets                              359623
In Unicast Packets                       7458
In Multicacast Packets                      0
In Broadcast Packets                        0
In Errors                                   0
In Discards                                 0
Signal Quality:
packets error-free                       8645
corrected                                   0
uncorrected                                 0
Signal Noise                              339  (dB)
Microreflections                           12

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Upstream Channel MAP Configuration

You can fine tune MAP generation for upstream channels, as described in the following sections.

Initial Maint Region Size (microsec)

The size of the upstream channel Initial Maintenance (IM) contention region. Maps with Initial Maint regions are sent periodically. By default, Initial Maint Contention Region Size is set at 500.

You set the initial maintenance contention for an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Configure the initial maintenance size for an upstream channel.	**upstream** <port number> **map init-maint-contention-size** <0..65535>

UCD Grant Size (microsec)

Upstream Channel Description (UCD) grant size. After a UCD change, this specifies the size of grant to zero (which functions as a delay for cable modems to digest the new UCD), in microseconds. By default, New UCD Grant Size is set at 3000.

Perform the following task within interface cable <c/s/i> mode to set the UCD grant size for an upstream channel.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Configure UCD grant size for an upstream channel.	**upstream** <port number> **map ucd-grant-size** <0..65535>

Maximum Deferred Ranging Invitations

Maximum number of deferred ranging invitations. By default, Maximum Deferred Ranging Invitations is set at 2.

You set maximum deferred ranging for an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Set maximum deferred ranging for a specific upstream channel.	**upstream** <port number> **map max-ranging-invitations** <0..65535>

Minimum Request Region

The minimum size, in minislots, for request contention region. By default, Minimum Request Region Size is set at 20.

You set maximum request contention region size for an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Set minimum request contention region for a specific upstream channel.	**upstream** <port number> **map min-request-region** <2-100>

Upstream Channel Ranging Configuration

You can fine tune how cable modems adjust power levels during the ranging process, as described in the following sections:

Power Offset Threshold (dB)

If power level offset reported by MAC chip is less than or equal to this threshold value, then power level adjustment may be stopped. By default, Power Offset Threshold is set at 8. Specified in 1/4 dB units.

You set the power offset threshold for an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Set the power offset threshold for the current cable interface.	**upstream** <port number> **ranging power-offset-threshold** <0..255>

CM Range Invite Timeout (millisec)

This is the minimum time allowed for a cable modem following receipt of a RNG-RSP, before it is expected to reply to an invitation to range request in milliseconds. By default, the CM Range Invite Timeout is set at 400 milliseconds.

You set the CM range invite timeout for an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Set CM range invite timeout.	**upstream** <port number> **ranging init-range-timeout** <0..65535>

Maximum Power Adjustment (1/4 dB)

The maximum adjustment permitted on a single Range Response message, specified in 1/4 dB units. By default, Maximum Power Adjustment is set at 6 dB.

You set the maximum power adjustment for an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Set maximum power adjustment.	**upstream** <port number> **ranging max-power-adjust** <0..255>

Enable Zero Power Adjustment

If enabled, the power adjustment field in the range response message is unconditionally set to 0. Useful for debugging. By default, Enable Zero Power Adjustment is disabled.

You enable or disable zero power adjustment on an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Enable or disable zero power adjustment.	**upstream** <port number> **ranging zero-power-adjust** {**enable** | **disable**}

Enable Zero Timing Adjustment

If enabled, the timing adjustment item in range response message is unconditionally set to 0. Useful for debugging. By default, Enable Zero Timing Adjustment is disabled.

You enable or disable zero timing adjustment on an upstream channel by performing the following tasks.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Enable or disable zero timing adjustment.	**upstream** <port number> **ranging zero-timing-adjust** {**enable** | **disable**}

Enable Zero Frequency Adjustment

If enabled, the frequency adjustment item in range response message is unconditionally set to 0. Useful for debugging. By default, Enable Zero Frequency Adjustment is disabled.

Perform the following task within interface cable <c/s/i> mode to enable or disable zero frequency adjustment on an upstream channel.

Task	Command
1. Enable interface cable mode.	**interface cable** <c/s/i>
2. Enable or disable zero frequency adjustment.	**upstream** <port number> **ranging zero-frequency-adjust** {**enable** | **disable**}

Configuring Admission Control

The admission control function allocates HFC interface bandwidth to service flows, and prevents admission of flows when bandwidth is unavailable. The admission control function sets aside bandwidth for unsolicited grant service (UGS) service flows and UGS with activity detection (UGS/AD) service flows, which are used to transmit voice traffic.

By default, admission control is disabled on an interface. Perform these tasks to configure admission control on a cable interface:

Task	Command
1. Access interface cable mode.	**interface cable** <c/s/i>
2. Enable admission control for the interface.	**admission-control enable**
3. Verify that admission control is enabled.	**show admission-control**
4. Reserve a percentage of bandwidth for UGS and UGS/AD service flows on one or more upstream ports. The <port number> argument specifies the upstream port for which you are reserving bandwidth. The **voice-bw-reserve** <number> argument specifies the percentage of bandwidth reserved. Values range from 0.0 to 100.0. The default is 75.0 percent.	**upstream** <port number> **voice-bw-reserve** <number>
5. Verify the bandwidth percentage setting.	**show upstream** <port number>

To disable admission control on a cable interface, perform these tasks:

Task	Command
1. Access interface cable mode.	**interface cable** <c/s/i>
2. Disable admission control.	**admission-control disable**
3. Verify that admission control is disabled.	**show admission-control**

Example

In this example, the administrator enables admission control on interface 1/1/1, and reserves a percentage of bandwidth on upstream 1.

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# admission-control enable
CMTS Admission Control:               Enable

cli:192.168.208.3:interface:cable:csi(1/1/1)# show admission-control
CMTS Admission Control:               Enable

cli:192.168.208.3:interface:cable:csi(1/1/1)# upstream 1 voice-bw-reserve 65
cli:192.168.208.3:interface:cable:csi(1/1/1)# show upstream 1
Upstream Channel Id        1 (1 / 1 / 3 / 2)
Center Frequency                     20.0  (MHz)
Channel Width                      3200.0  (KHz)
Slot Size                               2  (uSec)
Receive Power                           0  (TenthdBmV)
Voice BW Reservation                 65.0  (%)
Modulation Profile                      1
Tx Timing Offset                        0
Tx Backoff Start                        5
Tx Backoff End                         10
Ranging Backoff Start                   2
Ranging Backoff End                     3

Admin Status                           up
Operational Status                     up

Stats:
In Octets                        2752130
In Unicast Packets                  8540
In Multicast Packets                   0
In Broadcast Packets                8468
In Errors                              0
In Discards                            0
--More--
```

Configuring Frequency Hopping

This section describes frequency hopping, and includes the following sections:

- Understanding Frequency Hopping Configuration
- Understanding Frequency Hopping Parameters
- Frequency Hopping Statistics

Understanding Frequency Hopping Configuration

The ADC Policy-based Frequency Hopping function continuously monitors the quality of the upstream spectrum that is in use to avoid unacceptable error rates due to noise. When the plant quality degrades to an unacceptable level, the operating parameters of the tuned upstream are adjusted based on the policy configuration.

The quality of the channel is measured using spectrum quality indicators based on frame error rate. The frame error rate is determined by monitoring the pre and post Forward Error Correction (FEC) rates. The frame error rate is averaged over an amount of time and compared to a configured threshold. When the threshold is exceeded, the currently tuned upstream is considered a degraded spectrum and a decision is made based on the policy for this channel.

The Cuda 12000 allows you to configure a threshold that is used to determine when the upstream spectrum has degraded to an unacceptable level. This error threshold is a percentage of frames received in error in comparison to the total number of frames received not in error. If FEC is used, then frames in error is the number of pre and post FEC errors. If FEC is not being used, then the number of frames in error is the number of invalid frames received. This error threshold is averaged over an amount of time. When the error threshold is exceeded, the upstream spectrum is considered unacceptable and a change in the operating parameters for the channel is made based on the policies that are specified.

Understanding Frequency Hopping Parameters

Frequency Hopping parameters are set within interface:cable:csi <c/s/i> mode.

For each upstream channel you can configure up to five policies. Each of these policies consists of the following four parameters:

Table 18-3 Frequency Hopping Parameters

Parameter	Description
Threshold	The percentage error threshold for this frequency hopping policy entry.
Interval	The threshold interval for this frequency hopping policy entry in seconds.
Profile	The upstream burst profile number to be used when error threshold is reached in configured threshold interval time.
Frequency (MHz)	The center frequency value to be used when error threshold is reached in configured threshold interval time

When the error threshold is reached over the configurable time, the upstream frequency and burst profile is changed to those specified in the profile. This allows you to have considerable control and flexibility. For example, when upstream spectrum for a channel degrades, the initial policy specified may be to keep the tuned center frequency the same and increase FEC, or change from 16 QAM to QPSK in an attempt to improve the use of the spectrum. If the change in operating parameters based on the first policy fails to meet the configured error threshold, then the next policy may be to change to another center frequency in an attempt to find another channel with acceptable quality. Each of these policies is attempted in a round-robin fashion to maintain upstream quality.

To configure frequency hopping on an upstream channel, perform the following tasks in interface:cable:csi <c/s/i> mode:

Task	Command
1. Setup a policy for an upstream channel.	**spectrum-group** <1...5> **upstream** <port number>
2. Set the upstream burst profile number.	**spectrum-group** <rule number> **upstream** <port number> **profile** <upstream modulation number> {0...255}
3. Set the percentage error threshold.	**spectrum-group** <rule number> **upstream** <port number> **profile** <upstream modulation number> {0...255} **threshold** {1...100}
4. Set the threshold interval, in seconds.	**spectrum-group** <rule number> **upstream** <port number> **profile** <upstream modulation number> {0...255} **threshold** {1...100} **interval** {10...86400}
5. Set the center frequency value, in MHz	**spectrum-group** <rule number> **upstream** <port number> **profile** <upstream modulation number> {0...255} **threshold** {1...100} **interval** {10...86400} **frequency** {5.0...42.0}

Example

The following example configures Policy Number 1 on Upstream Channel 1:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# spectrum-group 1
upstream 1 profile 1 threshold 1 interval 10 frequency 5.0
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Display Policy Number 1 on all Upstream Channels

You display all Policy Number 1's on all upstream channels by performing the following tasks:

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Display the frequency hopping policy on an upstream channel.	**show spectrum-group** <rule number> **upstream**

Display All Policies on Single Upstream Channel

You display all policies on a single upstream channel by performing the following tasks:

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Display the frequency hopping policy on an upstream channel.	**show spectrum-group upstream** <port number>

Example

The following example displays all policies on upstream channel 1, and shows that there is only one policy configured.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show
spectrum-group 1 upstream 1
Upstream ID                                1
Rule Number                                1
Threshold                                  1
Interval                                  10
Frequency (MHz)                          5.0
Profile Num                                1

cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Display All Policies on All Upstream Channels

You display all policies on all upstream channels by performing the following tasks:

Task	Command
1. Enter interface cable mode.	**interface cable** <c/s/i>
2. Display the frequency hopping policy on all upstream channels.	**show spectrum-group upstream**

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show
spectrum-group upstream 1

row count: 3

Rule   Threshold Interval Frequency  Profile
Number                   (MHz)      Num
------ --------- -------- ---------- -------
     1         1       10        5.0       1
     2         1       10        5.0       1
     3         1       10        5.0       1

Current Rule: 1
cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

Frequency Hopping Statistics

Frequency Hopping also provides statistics for you to monitor the condition of your plant. Within **interface:cable:csi** mode you may view these statistics. You can display statistics for all policies on a single channel or all channels:

- **Error Rate**: The percentage of errors.
- **Error Count**: The number of frames with errors.
- **Total Packets**: The total number of frames received for each policy.

You display statistics for all policies on a single channel by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Display statistics to monitor the condition of your plant.	**show spectrum-group stats upstream** <port number>

You display statistics for all policies on all channels by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Display statistics to monitor the condition of your plant.	**show spectrum-group stats upstream**

Example

This example displays statistics for all policies on all channels:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show spectrum-group stats upstream
row count: 3

Rule   Upstream Threshold Interval Freq (MHz) Profile Error Error Count Error Packet
Number ID                                     Num     Rate              Count

------ -------- --------- -------- ---------- ------- ----- ----------- ------------
     1        1         1       10        5.0       1     0           0   1
     2        1         5       10       21.0       1     0           0   0
     2        2         5       10       21.0       1     0           0   0

cli:192.168.208.3:interface:cable:csi(1/1/1)#
```

***NOTE**: If you delete the current frequency hopping policy, then frequency hopping uses the next policy. If there is a gap between policies, then frequency hopping uses the next available policy.*

Defining Modulation Profiles

Modulation profiles contain the profile properties of the CMTS upstream data stream channels. The CMTS supports two profiles for the four upstream channels. Each profile consists of a burst descriptor for the following Interval Usage Codes:

- **Request**: Interval when a request on bandwidth can be sent by the modem.
- **Initial Maintenance**: Interval when new modems can start establishing a connection with CMTS with Initial Ranging Requests.
- **Station Maintenance**: Interval when modems perform periodic ranging with periodic ranging for adjusting power, timing and frequency.
- **Short Data**: Interval when a modem can send upstream PDU (Protocol Data Unit), which is less than one maximum burst size.
- **Long Data**: Interval when the modem can send upstream PDU, when one burst size exceeds one maximum burst size on the short data interval.

For each above-mentioned Usage Code, you must define eleven profile parameters as described in the following sections.

Caution: Profiles affect the physical layer. Changes to profile properties affects the performance and function of the CMTS. It is recommended that an expert-level user perform Modulation Profile configuration.

Guard Time

The Guard Time parameter is the number of symbol-times that must follow the end of this channel's burst. Note that this parameter is *read-only.*

Mod qpsk

The Mod qpsk parameter is the modulation type for an upstream channel. Acceptable values are QPSK and QAM16.

You set the modulation type by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Configure the modulation type.	**modulation-profile** <profile number> **interval-usage** {**initial** | **long** | **request** | **short** | **station**} **mod** {**qam16** | **qpsk**}

Pre-len

Preamble pattern length. You set the preamble length by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Configure the preamble length.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **pre-len** <2..448>

Burst-len

The Burst-len parameter is the maximum number of mini-slots that can be transmitted during a channel's burst time. A value of zero is transmitted if the burst length is bounded by the allocation MAP rather than this profile. By default, Max Burst Size is set to 0 for all interval usage codes.

You set the maximum burst size by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Configure the maximum burst size.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **burst-len** <0..255>

Diff

The Diff parameter enables differential encoding on this channel. By default, Differential Encoding is disabled. Differential Encoding should be enabled when FEC is not used and disabled when FEC is used.

You set the differential encoding by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Enable the differential encoding.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **diff**

Task	Command
3. Disable the differential encoding.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **no-diff**

FEC-tbytes

The Fec-tbytes parameter is the number of errored bytes that can be corrected in forward error correction code. By default, FEC-tbypes is set at zero. The value of zero indicates no correction is employed. Acceptable values are 0 to 10. The number of check bytes appended will be twice the value that is set.

You set the FEC-tbypes by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
Configure FEC error correction.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **fec-tbytes** <0..10>

Fec-len

The Fec-len parameter is the number of data bytes (k) in the forward error correction codeword. Acceptable values are 1 to 255. Note that this parameter is not used if FEC-tbytes is zero.

You set the codeword length by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Configure the codeword length.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **fec-len** <1...255>

Seed

The Seed parameter is the 15 bit seed value for the scrambler polynomial. By default, Seed is set to 338.

You set the scrambler seed by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Configure the scrambler seed value.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **seed** < 0x0000..0x7fff>

Shortened

The Shortened parameter enables the truncation of FEC codeword.

You specify whether to keep the codeword fixed or enable truncation of the FEC codeword by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Configure codeword length.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **last-cw** {**fixed** | **shortened**}

Scrambler

The Scrambler parameter enables or disables the scrambler.

You configure the scrambler by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Enable the scrambler.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **scrambler**
3. Disable the scrambler.	**modulation-profile** <profile number> **interval-usage** {**initial | long | request | short | station**} **no scrambler**

Example — Creating a Modulation Profile

The following example configures configures profile properties for all Usage Codes to create modulation-profile 3 on interface 1/1/1.

```
cli# interface 1/1/1
mode: interface:cable:csi(1/1/1)

cli# modulation-profile 3 interval-usage request fec-tbytes 6
cli# modulation-profile 3 interval-usage request fec-len 8
cli# modulation-profile 3 interval-usage request burst-len 120
cli# modulation-profile 3 interval-usage request mod qpsk

cli# modulation-profile 3 interval-usage request no scrambler
cli# modulation-profile 3 interval-usage request diff
cli# modulation-profile 3 interval-usage request seed 38
cli# modulation-profile 3 interval-usage request pre-len 8
cli# modulation-profile 3 interval-usage request last-cw fixed
cli# modulation-profile 3 interval-usage initial fec-tbytes 6
cli# modulation-profile 3 interval-usage initial fec-len 8
cli# modulation-profile 3 interval-usage initial burst-len 120
cli# modulation-profile 3 interval-usage initial mod qpsk

cli# modulation-profile 3 interval-usage initial no scrambler
cli# modulation-profile 3 interval-usage initial diff
cli# modulation-profile 3 interval-usage initial seed  38
cli# modulation-profile 3 interval-usage initial pre-len 8
cli# modulation-profile 3 interval-usage initial last-cw fixed

cli# modulation-profile 3 interval-usage station fec-tbytes 6
cli# modulation-profile 3 interval-usage station fec-len 8
cli# modulation-profile 3 interval-usage station burst-len 120
cli# modulation-profile 3 interval-usage station mod qpsk

cli# modulation-profile 3 interval-usage station no scrambler
cli# modulation-profile 3 interval-usage station diff
cli# modulation-profile 3 interval-usage station seed  38
cli# modulation-profile 3 interval-usage station pre-len 8
cli# modulation-profile 3 interval-usage station last-cw fixed

cli# modulation-profile 3 interval-usage short fec-tbytes 6
cli# modulation-profile 3 interval-usage short fec-len 8
cli# modulation-profile 3 interval-usage short burst-len 120
cli# modulation-profile 3 interval-usage short mod qpsk
cli# modulation-profile 3 interval-usage short no scrambler
```

```
cli# modulation-profile 3 interval-usage short diff
cli# modulation-profile 3 interval-usage short seed  38
cli# modulation-profile 3 interval-usage short pre-len 8
cli# modulation-profile 3 interval-usage short last-cw fixed

cli# modulation-profile 3 interval-usage long fec-tbytes 6
cli# modulation-profile 3 interval-usage long fec-len 8
cli# modulation-profile 3 interval-usage long burst-len 120
cli# modulation-profile 3 interval-usage long mod qpsk
cli# modulation-profile 3 interval-usage long no scrambler
cli# modulation-profile 3 interval-usage long diff
cli# modulation-profile 3 interval-usage long seed  38
cli# modulation-profile 3 interval-usage long pre-len 8
cli# modulation-profile 3 interval-usage long last-cw fixed
```

You can then verify the new modulation profile using the **show cable interface** <c/s/i> **modulation-profile** <profile number> command from within either root or cable interface mode, or the **show modulation-profile** <profile number> command from within cable interface mode.

Displaying Modulation Profiles

You display the modulation profiles currently configured on a cable interface by performing one of following tasks:

Task	Command
1. Enter this command within root mode or cable interface configuration mode to display all modulation profiles on the specified cable interface.	**show interface cable** <c/s/i> **modulation-profile**
2. Enter this command within cable interface configuration mode to display all modulation profiles on the specified cable interface.	**show modulation-profile**

Task	Command
3. Enter this command within root mode or cable interface configuration mode to display a specific modulation profile on the specified cable interface.	**show interface cable** <c/s/i> **modulation-profile** [<profile-index>]
4. Enter this command within cable interface configuration mode to display a specific modulation profile on the current cable interface.	**show modulation-profile** [<profile-index>]

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show modulation-profile 3

row count: 1

Interval Mod   Pream  Pream Diff FEC   FEC CW Scram Max   Guard Last Scram
Usage    Type  Offset Len        Error Len    Seed  Burst Time  CW   Enable
                                                    Size
-------- ----- ------ ----- ---- ----- ------ ----- ----- ----- ---- ------
Long     QPSK     488   384  yes     6      8     0   120     8  no    yes
```

Deleting Modulation Profiles

You delete a modulation profile from a cable interface by performing the following tasks

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Delete the specified modulation profile.	**no modulation-profile** <profile-index>

The following example deletes modulation profile 3 from interface 1/1/1:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show modulation-profile 3

row count: 1

Interval Mod   Pream  Pream Diff FEC   FEC CW Scram Max   Guard Last Scram
Usage    Type  Offset Len        Error Len    Seed  Burst Time  CW   Enable
                                                    Size
-------- ----- ------ ----- ---- ----- ------ ----- ----- ----- ---- ------
Short    QPSK     496   384  yes     5     34     0     0     8  no    yes

cli:172.16.19.10:interface:cable:csi(1/1/1)# no modulation-profile 3
cli:172.16.19.10:interface:cable:csi(1/1/1)# show modulation-profile 3
Modulation Profile 3 does not exist!
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Configuring CMTS Privacy Parameters

You can configure BPI and BPI+ privacy parameters on a cable interface. Information on configuring both CMTS and cable modem privacy parameters can be found in "Configuring BPI and BPI+ Parameters" onpage 453.

Configuring Flap Control

Flap Control configuration sets the control of the flap list by configuring the size and entry thresholds, as described in the following sections.

Flap-list size

The Flap-list size parameter sets the maximum number of entries (modems) in the flap list. By default, flat-list size sets the table at 8191 rows. Acceptable values are 0 to 8192.

You configure the maximum flap list size by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Set maximum flap table size.	**flap-list size** <value>

Flap-list aging

The Flap-list aging parameter sets the number of days to age the cable modem from the flap list table. Setting the aging threshold to zero results in modems never being aged from the table. By default, flap-list aging threshold is set at 60 days. Acceptable values are 1 to 60 days.

You configure the flap list aging threshold by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Set aging threshold.	**flap-list aging** <value>

Flap-list insertion-time

The Flap-list insertion-time parameter sets a threshold that controls the operation of a flapping modem detector. When the link establishment rate of a modem is shorter than the period defined by this parameter, the modem is placed in the flap list. Setting this parameter to zero results in modems never being inserted in the flap list table due to short link establishment times. By default, flap-list insertion-time Insert Time Threshold is set at 604800. Acceptable values are 0 to 604800 seconds.

You configure the flap list insertion time threshold by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Set flap list insertion threshold.	**flap-list insertion-time** <value>

Flap-list power-adj-threshold

The Flap-list power-adj-threshold parameter records a flap list event. When the power adjustment of a modem exceeds the threshold, the modem is placed in the flap list. Setting this parameter to zero results in modems never being inserted in the flap list table due to power adjustments. By default, the flap-list power-adj-threshold parameter is set at 3. Acceptable values are 1 to 10 dBmv.

You configure the power adjustment threshold by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Set flap list power adjustment threshold.	**flap-list power-adj-threshold** <value>

Show flap-list control

The Show flap-list control parameter displays all flap list control configuration, which you can use to verify the configuration.

Display the flap-list control information by performing the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Display flap list control configuration.	**show flap-list control**

For more information about flap lists, see "Managing Flap Lists" on page 466.

19 Managing Cable Modems

This chapter provides information on managing cable modems. Managing cable modems consists of the tasks listed below. These tasks do not need to be performed in specific order.

Viewing Cable Modems

Viewing cable modems consists of the following cable modem displays; all are entered using CLI commands.

- Displaying the Summary
- Displaying Detailed Listing
- Displaying a Specific Modem
- Displaying Cable Modem Statistics

Displaying the Summary of Cable Modem Registration States

You can display a summary of cable modems and their corresponding registration states. To do so, perform the following tasks:

Task	Command
1. Enter this command from within root mode or interface mode to display all cable modems on a specific interface.	**show interface cable** <c/s/i> **modem summary**
2. Enter this command from within cable interface mode to display all cable modems on the current interface.	**show modem summary**

Example

The following example displays the results of the **show modem summary** command. Refer to the next section, Displaying Detailed Listing, for explanations of the modem states.

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show modem summary

row count: 13


Modem Status Summary

Modem State            Ch1     Ch2     Ch3     Ch4     All
--------------------  ------  ------  ------  ------  --------
DhcpReqRcvd                 0       1       0       0         1
Registered                  6       4       0       0        10
TimeReqRcvd                 1       1       0       0         2
--------------------  ------  ------  ------  ------  --------
Total                       7       6       0       0        13

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Displaying a Detailed Listing for an Interface

You can display a list of cable modems attached to a specific cable interface and their associated status. To do so, perform the following tasks:

Task	Command
1. Enter this command from within root mode or interface mode to display all cable modems on a specific interface.	**show interface cable** <c/s/i> **modem**
2. Enter this command from within cable interface mode to display all cable modems on the current interface.	**show modem**

Example

The following example shows the result of the **show modem** command within cable interface configuration mode to display a list of attached modems:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:96:00:29:71 201.1.1.112       1    1   0 1:2      0   2216 Registered
00:90:83:36:82:ee 201.1.1.108       2    0   0 1:2      0   1240 RegBpiTek
00:90:96:00:29:6d 201.1.1.104       3    1   0 1:2      0   2216 Registered
00:10:95:04:0a:b7 201.1.1.109       4    1   0 1:2      0   2726 Registered
00:90:96:00:39:f9 201.1.1.101       5    1   0 1:2      0   2212 Registered
00:10:95:01:ef:d8 201.1.1.100       6    1   0 1:2      0   2208 RegFailBad
00:10:95:04:0a:c3 0.0.0.0           7   57   0 1:2     -9   2724 Ranging
00:90:96:00:39:7f 201.1.1.102       8    1   0 1:2      0   2219 Registered
00:10:95:04:0a:c4 201.1.1.110       9    1   0 1:2      0   2724 Registered
```

The display includes the following parameter information:

Table 19-1 Cable Modem Display Parameters

Parameter	Description
MAC Address	The RF MAC address of this cable modem.
IP Address	The IP address assigned to this cable modem by DHCP.
SID	The service ID assigned to this cable modem.
CID	DOCSIS 1.0 class of service ID.
CPE	The number of CPE devices attached to this cable modem.
D:U	Downstream and Upstream channel IDs. D: The downstream channel assigned to the cable modem. The Cuda 12000 supports only one downstream channel. U: The upstream channel assigned to the cable modem. The Cuda 12000 supports up to 6 upstream channels.
Power (dbMV)	The receive power as perceived for upstream data from this cable modem. If the receive power is unknown, zero is displayed.

Timing	Timing Offset. A measure of the current round-trip time for this cable modem. Timing Offset is used for timing cable modem upstream transmissions to ensure synchronized arrivals at the CMTS. Units are in terms of (6.25 microseconds/64). A value of zero is returned if the time is unknown.
Modem State	Current cable modem connectivity state specified in the DOCSIS 1.0 and 1.1 RF Interface Specifications. Returned status information is the cable modem status as assumed by the CMTS. The possible status values include: **InitRngRcvd**: The CMTS received an Initial Ranging Request message from the cable modem and the ranging process is not yet complete. **Ranging**: The modem is in the process of ranging. **RangingComplete**: The CMTS sent a Range Response (success) message to the cable modem. **DhcpDiscRcvd**: The CMTS has received a DHCP Discover message from the cable modem. **DhcpReqRcvd**: The CMTS has received a DHCP Request from the cable modem. **TimeReqRcvd**: The CMTS has received a Time Request.

Modem State (continued)	**TftpReqRcvd**: The CMTS has received a TFTP Request from the cable modem. **Registered**: The cable modem is registered, without Baseline Privacy. **RegNoNetAccess**: The cable modem is registered, but Network Access is disabled. **RegBpiKek**: The cable modem is registered, with Baseline Privacy enable. A Key Encryption Key has been assigned. **RegBpiTek**: The DOCSIS 1.1 cable modem is registered, with Baseline Privacy enabled. A Traffic Encryption Key has been assigned. **RegFailBadMic**: Modem registration failed, due to CMTS MIC comparison failure. There is a shared key mismatch. **RegFailBadCos**: Modem registration failed, due to class of service failure. **RegFailAuth**: Modem registration failed, due to authorization failure. **RegKekReject**: The cable modem is registered, with Baseline Privacy enabled. A Key Encryption Key has been rejected**.** **RegTekReject**: The cable modem registered, with Baseline Privacy enabled. A Traffic Encryption Key has been rejected.

Displaying Specific Cable Modems

You can display selected modems by MAC address. To do so, perform the following tasks:

Task	Command
1. Enter this command within root mode or interface mode to display a modem for a specified interface.	**show interface cable** <c/s/i> **modem** <mac-address>
2. Enter this command within interface mode to display a specified modem connected to the current cable interface.	**show modem** <mac-address>

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem 00:90:83:32:9f:8c
S l o t                                  1
MAC Address                00:90:83:32:9f:8c
IP Address                       201.1.1.110
SID                                     12
CID                                      1
CPE                                      0
D:U                                    1:2
Power                                    0  (dbMV)
Timing                                1652
Modem State                     Registered
```

Displaying Cable Modem Statistics

To display cable modem statistics, perform the following tasks:

Task	Command
1. Enter this command from within **root** mode or cable **interface** mode to display statistics for cable modems attached to the specified interface.	**show interface cable** <c/s/i> **modem stats**
2. Enter this command within **cable interface** mode to display statistics for cable modems attached to the current cable interface.	**show modem stats**

Example

The following example shows the results of the **show modem stats** command.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem stats

row count: 12

MAC Address       IP Address      Vendor  Pkts NonErr    CorrErr   UnCorr
                                  Name
----------------- --------------- ------- ---- --------- --------- -------
00:90:96:00:29:71 201.1.1.102     ASKEY     22       638         0       0
                                  COMPUTE
00:90:96:00:29:6d 201.1.1.103     ASKEY     22       638         0       0
                                  COMPUTE
00:10:95:01:f0:05 201.1.1.100     THOMSON   28       636         0       0
                                  CONSU
00:10:95:04:0a:c3 0.0.0.0         THOMSON    0     23463         0       0
                                  CONSU
00:10:95:04:0a:b7 201.1.1.101     THOMSON   54       636         0       0
                                  CONSU
00:90:96:00:39:7f 201.1.1.107     ASKEY     22       639         0       0
                                  COMPUTE
00:90:96:00:39:f9 201.1.1.105     ASKEY     22       638         0       0
                                  COMPUTE
00:10:95:04:0a:c4 201.1.1.104     THOMSON   54       635         0       0
                                  CONSU
00:10:95:01:ef:d8 201.1.1.106     THOMSON   28       635         0       0
```

Tracking Offline Cable Modems

You can control how long the CMTS tracks offline cable modems, and manage cable modem statistics when a cable modem transitions out of the offline state.

Tracking offline cable modems involves:

- Specifying the number of days that you want the CMTS to track offline cable modems.
- Specifying whether you want the CMTS to maintain cable modem statistics when the cable modem transitions out of offline state.
- Managing offline cable modems.

Setting the Duration for Tracking Offline Cable Modems

To specify the number of days that you want the CMTS to track offline cable modems, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Specify the duration of time, in days, that you want the CMTS to track offline cable modems. Values range from 0 to 365. The default is 30.	**cm-offline timer** <integer>

Example

The following example sets the offline timer to 35 days:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# cm-offline timer 35
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-offline
Cable Modem Offline Timer                35
Cable Modem Stats Persist             enabled
```

Maintaining Statistics for Offline Cable Modems

To specify whether you want the CMTS to maintain the statistics when a cable modem transitions out of the offline state, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Specify that you want the CMTS to maintain the statistics.	**cm-offline persist**
3. Specify that you do not want the CMTS to maintain the statistics.	**no cm-offline persist**

Example

The following example sets the CMTS to persist statistics when a cable modem transitions out of offline state:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# cm-offline persist
cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-offline
Cable Modem Offline Timer                30
Cable Modem Stats Persist              enabled
```

Clearing Offline Cable Modems

You can set the CMTS to remove all offline cable modems from the summary table. To remove all offline cable modems, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Set the CMTS to remove all offline cable modems from the summary table.	**cm-offline clear**

Resetting Cable Modems

The Cuda lets you reset a single modem or multiple modems attached to the same cable interface. You can specify the modem that you want to reset in terms of its IP address, MAC address, or Service Identifier (SID).

Use the **cm reset** command within interface cable <c/s/i> mode to reset cable modems.

Resetting a Single Modem

To reset a single modem, perform the following task within interface cable <c/s/i> mode:

Task	Command
Reset a single modem	**cm reset** {<ip-address> | <mac-address> | <sid>}

Example

The following example shows the results of the **cm reset** command when you reset a single modem with the MAC address 00:90:83:31:ac:ad:

```
cli:172.16.19.10:root# interface cable 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# cm reset 00:90:83:32:9f:8c
Reseting Cable Modem4316 with Mac Address: 00:90:83:32:9f:8c
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Resetting Multiple Modems

To reset multiple modems with common MAC address hexadecimal values, perform the following task within interface cable <c/s/i> mode.

Task	Command
Reset multiple modems with common hexidecimal values.	**cm reset** <hex-values> **match** The hex values that you want to match entered with "ff" values specified as wildcards. For example, if you want to reset all cable modems from vendor 00:50:72, enter the following command: **cm-reset 00:50:72:ff:ff:ff match**

Example

The following example displays the modems attached to cable interface 1/1/1. The administrator uses the **match** argument to reset all modems with the vendor ID: 00:90:96.

```
cli:interface:cable:csi(1/1/1)# show modem

row count: 11

MAC Address       IP Address      SID  CID   CPE   D:U Power  Timing Modem
                                                     (dbMV)        State
----------------- --------------- ---- ----  ---   --- ------ ------ -------
00:90:96:00:29:6d 201.1.1.103      453    1    0   1:2      0   2217 Registered
00:10:95:01:ef:d8 201.1.1.100      654    1    0   1:2      0   2209 Registered
00:10:95:04:0a:c4 201.1.1.102       63    1    0   1:2      0   2729 Registered
00:90:96:00:39:f9 201.1.1.101       68    1    0   1:2      0   2220 Registered
00:10:95:04:0a:b7 201.1.1.105       64    1    0   1:2      0   2727 Registered
00:90:96:00:29:71 201.1.1.104       66    1    0   1:2      0   2222 Registered
00:10:95:04:0a:bd 201.1.1.106       61    1    0   1:2     -2   2729 Registered
00:10:95:01:f0:05 201.1.1.107       75    1    0   1:2      0   2214 Registered
00:90:83:36:82:f1 201.1.1.110       73    1    0   1:2      0   1234 Registered
00:90:83:32:9f:8c 201.1.1.111       70    1    0   1:2      0   1657 Registered
00:10:95:04:0a:c3 0.0.0.0           62    0    0   1:2     -8   2730 Ranging

cli:interface:cable:csi(1/1/1)# cm reset 00:90:96:ff:ff:ff match
Resetting Cable Modem with Mac Address: 00:90:96:00:29:6d
Resetting Cable Modem with Mac Address: 00:90:96:00:39:f9
Resetting Cable Modem with Mac Address: 00:90:96:00:29:71
```

Resetting All Modems on a Network

To reset all modems attached to a selected network, perform the following task within the interface cable <c/s/i> mode.

Task	Command
Reset all modems on a network.	**cm reset <**address-string> **match** The IP address that you want to match entered with a "255" wildcard mask. For example, if you want to reset all cable modems attached to subnet 189.23.3.x, enter the following command: **cm reset 189.23.3.255 match**

Example

The following example uses the match argument against the IP address parameter to reset all cable modems on the 192.168.19.x subnet.

```
cli:172.16.19.10:root# interface cable 1/1/1
mode: interface:cable:csi(1/1/1)
cli:172.16.19.10:interface:cable:csi(1/1/1)# cm reset 192.168.255.255 match
Reseting Cable Modem4305 with Ip Address: 192.168.19.52
Reseting Cable Modem4307 with Ip Address: 192.168.19.53
Reseting Cable Modem4308 with Ip Address: 192.168.19.55
Reseting Cable Modem4309 with Ip Address: 192.168.19.59
Reseting Cable Modem4310 with Ip Address: 192.168.19.58
Reseting Cable Modem4311 with Ip Address: 192.168.19.56
Reseting Cable Modem4312 with Ip Address: 192.168.19.57
Reseting Cable Modem4313 with Ip Address: 192.168.19.60
Reseting Cable Modem4314 with Ip Address: 192.168.19.54
Reseting Cable Modem4315 with Ip Address: 192.168.19.51
Reseting Cable Modem4318 with Ip Address: 192.168.19.63
Reseting Cable Modem4319 with Ip Address: 192.168.19.61
Reseting Cable Modem4320 with Ip Address: 192.168.19.62
cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Changing Upstream Channels

You can change the upstream channel that a cable modem is using by changing the upstream channel ID for the specified modem.

To change the upstream channel, perform the following task in interface cable <c/s/i> mode.

Task	Command
Modify the upstream channel ID for a select modem.	**cm modify upstream** <new-upstream-channel> {<ip-address> | <mac-address> | <sid>}

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:96:00:29:71 201.1.1.102       1    1   0 1:2      0   2215 Registered
00:90:96:00:29:6d 201.1.1.103       2    1   0 1:2      0   2217 Registered
00:10:95:01:f0:05 201.1.1.100       3    1   0 1:2      0   2210 Registered
00:10:95:04:0a:c3 0.0.0.0           4    0   0 1:2     -9   2723 Ranging
00:10:95:04:0a:b7 201.1.1.101       5    1   0 1:2      0   2726 Registered
00:90:96:00:39:7f 201.1.1.107       6    1   0 1:2      0   2216 Registered
00:90:96:00:39:f9 201.1.1.105       7    1   0 1:2      0   2212 Registered
00:10:95:04:0a:c4 201.1.1.104       8    1   0 1:2      0   2727 Registered
00:10:95:01:ef:d8 201.1.1.106       9    1   0 1:2      0   2205 Registered
00:90:83:36:82:f1 201.1.1.108      10    1   0 1:2      0   1247 Registered
00:90:83:36:82:ee 201.1.1.109      11    1   0 1:2      0   1228 Registered
00:90:83:32:9f:8c 201.1.1.110      12    1   0 1:2      0   1652 Registered

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify upstream 1 2
Modifying Cable Modem with SID: 2 to upstream channel: 1
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem 00:90:96:00:29:71
S l o t                                1
MAC Address               00:90:96:00:29:71
IP Address                      201.1.1.102
SID                                    1
CID                                    1
CPE                                    0
D:U                                  1:2
Power                                  0  (dbMV)
Timing                              2214
Modem State                   Registered
```

Viewing Services

Services are assigned when the cable modems are provisioned. The CMTS dynamically assigns a Service ID number to the cable modem. A cable modem keeps the same Service ID for as long as it continues to range and is registered with the CMTS. For example, if a cable modem is reset or goes through a power cycle, CMTS reassigns the next available Service ID number to the cable modem the next time it ranges and registers. To view a list of services currently assigned by the cable interface, perform the following task in interface cable <c/s/i> mode.

Task	Command
1. Enter this command from within root mode or cable interface configuration mode to sort the cable modem display according to services assignment ID.	**show interface cable** <c/s/i> <sid>
2. Enter this command from within cable interface configuration mode to sort the cable modem display according to services assignment ID.	**show** <sid>
3. Issue this command from within root mode or cable interface configuration mode to display a specific cable modem according to SID.	**show interface cable** <c/s/i> **sid** <sid>
4. Enter this command from within cable interface configuration mode to display a specific cable modem according to SID.	**show sid** <sid>

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show sid
row count: 12
SID   QOS    Create Time       Class MAC Address       IP Address
                               ID
----- ------ ----------------- ----- ----------------- ---------------
    1   1025 01-09-17 15:36:43     1 00:90:96:00:29:71 201.1.1.102
    2   1025 01-09-17 18:56:12     1 00:90:96:00:29:6d 201.1.1.103
    3   1025 01-09-17 15:36:42     1 00:10:95:01:f0:05 201.1.1.100
    4   1026 01-09-17 15:36:41     0 00:10:95:04:0a:c3 0.0.0.0
    5   1025 01-09-17 15:36:25     1 00:10:95:04:0a:b7 201.1.1.101
    6   1025 01-09-17 15:36:41     1 00:90:96:00:39:7f 201.1.1.107
    7   1025 01-09-17 15:36:46     1 00:90:96:00:39:f9 201.1.1.105
    8   1025 01-09-17 15:36:39     1 00:10:95:04:0a:c4 201.1.1.104
    9   1025 01-09-17 15:36:48     1 00:10:95:01:ef:d8 201.1.1.106
```

The SID summary display includes the following parameter information:

Table 19-2 SID Summary Display Parameters

Parameter	Description
SID	The Service Id number assigned dynamically to the cable modem by the CMTS.
QoS	The QoS Profile provisioned to this cable modem.
Create Time	The date and time at which the SID was assigned to the cable modem.
Class ID	The DOCSIS 1.0 class of service ID.
MAC Address	MAC address of the cable modem to which this SID is assigned.
IP Address	IP address of the cable modem to which this SID is assigned.

To view statistics for each service identifier, perform the following task in interface cable <c/s/i> mode.

Task	Command
View SID statistics.	**show interface cable** <c/s/i> **sid stats**

Example

The following example displays the results of the **show sid stats** command.

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show sid stats

row count: 13

SID   In Pkts     In Disc     Out Pkts    Out Disc    Bw Reqs     Bw Grants
----- ---------- ---------- ---------- ---------- ---------- ----------
  669         10          0          5          0         14         13
  670         10          0          5          0         12         12
  671         10          0          5          0         11         11
  672         10          0          5          0         11         11
  673         11          0          3          0         12         12
  674         11          0          3          0         12         12
  675         11          0          3          0         12         12
  676         11          0          3          0         12         12
  677         22          0         16          0         23         23
  678         12          0          6          0         14         14
  679         20          0         14          0         22         21
  680         20          0         14          0         22         21
  681         12          0          6          0         14         13

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The SID statistics display includes the following parameter information:

Table 19-3 SID Statistics

Parameter	Description
SID	The Service Id number assigned dynamically to the cable modem by the CMTS.
In Pkts	The number of packets received by this cable modem.
In Disc	The aggregate number of discard packets received.
Out Pkts	The number of packets transmitted from this cable modem
Out Disc	The aggregate number of discard packets transmitted
BW Reqs	The number of bandwidth requests received from this cable modem.
BW Grants	The number of bandwidth requests transmitted to this cable modem.

Configuring BPI and BPI+ Parameters

Configuring BPI and BPI+ includes the following tasks:

- Configuring Authorization and Traffic Encryption Keys. This task applies to both BPI and BPI+.
- Configuring Trust and Validity for Manufacturer Certificates. This task applies to BPI+ only.
- Configuring IP Multicast Address Mapping. This task applies to BPI+ only.
- Viewing privacy keys. This task applies to both BPI and BPI+.

About BPI and BPI Plus

The Baseline Privacy Interface (BPI) and BPI Plus (BPI+) protocols provide cable modems with data privacy across the Hybrid Fiber-Coaxial (HFC) network by encrypting traffic between cable modems and the CMTS.

NOTE: *For a cable modem to use BPI and BPI+, you must configure the Baseline Privacy settings in the modem configuration file. This file downloads during the transfer of operation parameters.You create configuration files within the cable modem provisioning process. Refer to the FastFlow Broadband Provisioning Manager CLI-based Administration Guide, or the guide of your third party provisioning manager.*

Both BPI and BPI+ provide authorization parameters and traffic encryption keys that secure traffic between cable modems and the CMTS. In addition, BPI+ provides:

- Authentication of cable modems through digital certificates. A cable modem can use a digital signature to verify that the software image it has downloaded has not been altered or corrupted in transit.
- Strong protection from theft of service. The protected communication falls into three categories:
  - Best effort, high speed, IP data services
  - Quality of Service (QoS)
  - IP multicast group services.

During the CMTS registration process, the CMTS assigns one or more static Service Identifiers (SIDs) to the registering cable modem that matches the cable modems class-of-service provisioning. The first static SID that the CMTS assigns is the primary SID and serves as the cable modem's primary Security Association Identifier (SAID). If the cable modem is configured to operate BPI+, the cable modem's BPI+ security functions initialize.

After successfully completing authentication and authorization with the CMTS, the cable modem sends a request to the CMTS requesting Traffic Encryption Keys (TEKs) to use with each of the SAIDs. The CMTS response contains the TEKs.

Configuring Authorization and Traffic Encryption Keys

You can configure and display lifetime in seconds for all new authorization and Traffic Encryption Keys (TEK), as well as for existing authorization and TEKs for a specified interface or a specified cable modem. Note that this task applies to both BPI and BPI+.

Configuring lifetime for authorization keys and TEK involves setting the following parameters:

Table 19-4 Parameters for Setting and Displaying Lifetime in Seconds

Parameter	Description
interface cable <c/s/i>	The interface for which you want to configure and display lifetime for authorization keys and TEK.
mac-address	The MAC address of the cable modem to which the CMTS assigns an authorization key and TEK.
invalidateAuth	The CMTS invalidates the current cable modem authorization keys, but does not transmit an authorization message or invalidate unicast TEKs.
sendAuthInvalid	The CMTS invalidates the current cable modem authorization key and transmits an invalid message to the cable modem. The CMTS does not invalidate the unicast TEKs.
invalidateTeks	The CMTS invalidates the current authorization key and transmits an authorization invalid message to the cable modem. The CMTS also invalidates all unicast TEKs associated with this cable modem authorization.
SAID	The DOCSIS 1.1 Baseline Privacy security identifier between the CMTS and the cable modem. The range of identifier values is 0..4294967295.

Parameter	Description
stats	Displays the statistics of the BPI+ configuration.
tek-lifetime	Specifies the allowable value range for the TEK lifetime. Values range from 1 to 6048000 seconds.
40-bit-des	Configures the interface for 40-bit baseline privacy encryption.
56-bit-des	Configures the interface for 56-bit baseline privacy encryption.

You use the following commands within interface cable <c/s/i> mode to configure and display lifetime in seconds:

Task	Command
1. Set the lifetime in seconds that the CMTS assigns to a new authorization key.	**privacy base auth-lifetime** <1..6048000>. The default value is 6048000.
2. Set the lifetime in seconds that the CMTS assigns to a new TEK.	**privacy base tek-lifetime** <1..604800>. The default value is 43200.
3. Set the lifetime in seconds that the CMTS assigns to an authorization key for a specified cable modem.	**privacy auth** <mac-address> {**cm-lifetime** <1..6048000> | **cm-reset** {**invalidateAuth** | **invalidateTeks** | **sendAuthInvalid**}}. The default value for **cm-lifetime** is 6048000.
4. Set the lifetime in seconds that the CMTS assigns to a TEK for an associated Security Association Identifier (SAID).	**privacy tek** <said> {**tek-lifetime** <1..6048000> | **reset**} The default tek-lifetime is 43200.
5. Set the type of encryption to be used for encrypting keys on the interface.	**privacy encryption** {**40-bit-des** | **56-bit-des**}

Task	Command
6. Display the current BPI and BPI+ base configuration, for all interfaces or for a specified interface.	**show** [**interface cable** <c/s/i>] **privacy base**
7. Display the authorization key configuration and statistics for all interfaces, a specified interface, or a specified cable modem.	**show** [**interface cable** <c/s/i>] **privacy auth** [<mac-address>] {**stats | error**}
8. Display the TEK configuration and statistics with associated SAID for all interfaces, or a specified interface.	**show** [**interface cable** <c/s/i>] **privacy tek** [<said>] [**stats**]

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy base auth-lifetime 43000
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy base tek-lifetime 50000
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy tek 8192 tek-lifetime 604800
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy encryption 56-bit-des
```

Configuring Trust and Validity for Manufacturer Certificates

A Certificate Authority (CA) is a self-signed certificate containing the DOCSIS CA's trusted public key. The manufacturer issues an X.509 certificate that binds the cable modem public key to other identifying information. BPI+ uses the X.509 digital certificate to authenticate key exchanges between the cable modem and CMTS.

You can configure and display trust for all new self-assigned manufacturer certificates, as well as for existing certificates for a specified cable modem. In addition you can configure certificates to have or not to have their validity period checked against the current time of day. Note that this task applies to BPI+ only.

Configuring trust and validity for certificates involves setting the following parameters:

Table 19-5 Parameters for Setting and Displaying Trust and Validity for Certificates

Parameter	Description
interface cable <c/s/i>	The interface for which you want to display certificates.
mac-address	The MAC address of the cable modem for which you want to display certificates.
trusted	Sets a valid certificate.
untrusted	Sets an invalid certificate. The default is set to untrusted.
enable	Sets the certificate to True. This means that the validity is checked against the current time of day.
disable	Sets the certificate to False. This means that the validity is not checked against the current time of day.
learnt	Indicates that you want to display the certificates for the cable modems.
provisioned	Indicates that you want to display the certificates for the provisioned cable modem.
details	Indicates that you want to display the BPI+ privacy authorization for the provisioned cable modem.

Parameter	Description
filename	The CM Configuration file name associated to the certificate. These certificates reside in the following directory in Linux on the Cuda 12000: **/bas/data/certification**
chained	Specifies that the certificate's level of trust is chained. In order for a chained certificate to be valid, it must meet several criteria, such as: ■ The certificate is linked to a Root, Trusted, or Valid certificate. ■ The certificate's signature can be verified with the issuer's public key. ■ The current time falls within the validity period of each Chained or Root certificate within the certificate chain.
root	Specifies that the certificate's level of trust is root. Note that only the DOCSIS Root CA Certificate (a self-signed certificate containing the DOCSIS Root CA's trusted public key) must be marked as Root. However, a CMTS MAY support multiple Root CA Certificates. At least one root certificate must be provisioned.

Before you configure trust and validity for certificates, you must create the **/bas/data/certification** directory in Linux on the Cuda 12000. Certificates will reside in this directory. To create this directory, access Linux on the Cuda 12000 through the local console or an SSH session with sufficient access privileges (such as root). Then, use the Linux **mkdir** command to create the directory.

You use the following commands to configure and display trust and validity for certificates. Note that all of these commands are issued from within interface cable <c/s/i> mode, except for **privacy ca-cert** and **privacy cm-cert**. These two commands can be issued from within either interface cable <c/s/i> mode or root mode.

Task	Command
1. Set the trust for all new self-assigned manufacturer certificates.	**privacy base cert-trust** {**trusted** | **untrusted**}.
2. Set the certificates to have (true) or not to have (false) the validity period checked against the current time of day.	**privacy base enable-cert-validity-periods** {**true** | **false**}
3. Display the certificate settings for the specified cable modem.	**show** [**interface cable** <c/s/i>] **privacy cm-cert** [<mac-address>] {**learnt | provisioned** [**details**]}.
4. Assigns a certificate to a provisioned cable modem.	**privacy cm-cert** <mac-address> [{**trusted** | **untrusted**}] **certificate** <filename>
5. Remove the certificate assignment for the provisioned cable modem.	**no privacy cm-cert** <mac-address>.
6. Specify manufacturer certification authority certificates.	**privacy ca-cert** <1..10000> {**trusted** | **untrusted** | **chained** | **root**} **certificate** <filename>
7. Remove the manufacturer certification authority certificate.	**no privacy ca-cert** <1..10000>

Example

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy base cert-trust trusted
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy base
enable-cert-validity-periods true
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy cm-cert 00:10:95:04:0a:c3
trusted certificate cm64.cer
cli:192.168.208.3:interface:cable:csi(1/1/1)# privacy ca-cert 5 trusted
certificate manf64.cer
```

Configuring IP Multicast Address Mapping

You can configure and display an IP multicast address mapping entry, and set the associated multicast SAID authorization for each cable modem on each CMTS MAC interface.

Configuring an IP multicast address mapping and associated SAID involves setting the following parameters:

Table 19-6 Parameters for Setting and Displaying IP Multicast Address Mappings

Parameter	Description
index	An index value that identifies the multicast mapping entry.
multicast-ip-address	Specifies the Class D IP address (for example, 239.1.1.1) of the multicast group to which you are applying the security association specified by the SAID.
mask	Specifies the mask, in dot-notation format, that can be used with a single multicast group address or to specify a multicast address range. For example, a multicast address of 239.1.0.0 and a mask of 255.255.0.0 means that the SA applies to all multicast groups within 239.1.0.0. For a single multicast group address (for example, 239.2.2.2), specify a mask of 255.255.255.255.
mac-address	The MAC address of the CM that is authorized to access the defined multicast stream via the SAID.
SAID	The multicast SAID used in the multicast address mapping entry. Allowable value ranges are 8192 to 16383.

Parameter	Description
sa-type	Specifies one of the following security association types: ■ dynamic – Specifies a Dynamic Security Association, which is established and eliminated on the fly in response to the initiation and termination of specific (downstream) traffic flows. Both Static and Dynamic SAs can by shared by multiple CMs. ■ none – Specifies no security association. ■ primary – Specifies Primary Security Association, which is tied to a single cable modem, and is established when that cable modem completes DOCSIS MAC registration. ■ static – Specifies a Static Security Association, which is provisioned within the CMTS.
encrypt-alg	Specifies one of the following encryption algorithms: ■ des40cbcMode – Specifies 40-bit DES packet data encryption. ■ des56cbcMode – Specifies 56-bit DES packet data encryption. ■ none – Specifies no encryption.
authent-alg	At this time, only a value of **none** is supported.

You use the following commands within interface cable <c/s/i> mode to configure and display IP multicast address mapping entries:

Task	Command
1. Set the IP multicast address mapping entry and its mask for the associated SAID.	**privacy multicast ip** <index> <multicast-ip-address> <mask> **said** <number> **sa-type** {**dynamic | none | primary | static**} **encrypt-alg** {**des40cbcMode | des56cbcMode | none**} **authent-alg none**
2. Remove the IP multicast address mapping entry.	**no privacy multicast ip** <index>
3. Set the MAC address of the CM that is authorized to access the defined multicast stream via the SAID.	**privacy multicast mac** <mac-address> **said** <8192..16383>
4. Remove the multicast MAC entry.	**no privacy multicast mac** <mac-address> **said** <8192..16383>
5. Display IP and MAC multicast address entries.	**show** [**interface cable** <c/s/i>] **privacy multicast** {**ip** [<index>] | **mac** [<mac-address>]}

Example

```
cli:interface:cable:csi(1/1/1)# privacy multicast ip 1 239.2.2.2 255.255.255.255
said 8192 sa-type dynamic encrypt-alg des56cbcMode authent-alg none
cli:interface:cable:csi(1/1/1)# show privacy multicast ip

row count: 1

Index IP Address      IP Mask         SAID  SA Type Encrypt Alg  Authent
                                                                 Alg
----- --------------- --------------- ----- ------- ------------ -------
    1 239.2.2.2       255.255.255.255  8192 dynamic des56cbcMode    none

cli:interface:cable:csi(1/1/1)# privacy multicast mac 00:10:95:04:0a:c3 said 8192
```

Viewing Privacy Keys

To display privacy key (that is, TEK) information, perform the following task within interface:cable:csi <c/s/i> mode:

Task	Command
Display privacy key information.	**show** [**interface cable** <c/s/i>] **privacy tek** [**said**] [**stats**]

When issued without the optional **said** and **stats** arguments, the display includes the following parameter information:

Table 19-7 Privacy Parameters

Parameter	Description
SAID	The security association ID.
SA Type	Displays the security association type (dynamic, none, primary, or static). Refer to "Configuring IP Multicast Address Mapping" on page 461 for more information on these types.
Encryption Algorithm	Displays the algorithm used to encrypt the key.
Authentication Algorithm	No authentication algorithm is supported at this time.
Life Time	Displays the lifetime of the key, in seconds.
Tek Reset	Specifies "true" if the TEK value range is reset. Otherwise, this field displays "false."

Sequence Number	Displays the authorization sequence number assigned to the key.

When issued with the **said** argument, the display shows the same information, but for the specified SAID only.

When issued with the **stats** argument, the display shows the following statistics:

Table 19-8 Privacy Statistics

Statistic	Description
SAID	The security association ID.
Requests	The number of privacy key requests received by the CMTS.
Replies	The number of privacy key replies sent by the CMTS.
Rejects	The number of Authorization Reject messages sent by the CMTS.
Invalids	The number of Authorization Invalid messages sent by the CMTS.

Managing Flap Lists

The flap list monitors the cable modems that have connectivity problems. Flapping refers to the rapid disconnecting and reconnecting of a cable modem experiencing problems holding a connection.

The function of the flap list includes:

- Maintaining entries for cable modems that completed registration and subsequently reset.
- Logging the time of the most recent activity of the cable modem by MAC address.

Viewing the Flap List

To display the flap list for a specific cable interface, perform the following tasks:

Task	Command
1. Display the flap list for a specified cable interface.	**show interface cable** <c/s/i> **flap-list**
or	
1. Sort the flap-list display by flap count.	**show interface cable** <c/s/i> **flap-list sortbyflapcnt**
2. Sort the flap-list display by time.	**show interface cable** <c/s/i> **flap-list sortbytime**

Example

The following example displays the flap list for cable interface 1/1/1:

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show flap-list

row count: 13

Mac Address       Flap  Last Known    Insert Time    Remove Time    Hit   Miss
                  Count State                                     Count Count
----------------- ----- ------------- -------------- -------------- ----- -----
00:10:95:01:ef:d8    19 Registered    99-09-03 19:06 99-09-03 19:06 10998   256
00:10:95:01:f0:05    16 Registered    99-09-03 19:06 99-09-03 19:06 10979   208
00:10:95:04:0a:b7    22 Registered    99-09-03 19:06 99-09-03 19:06 11031   304
00:10:95:04:0a:c3    22 Registered    99-09-03 19:06 99-09-03 19:06 11026   304
00:10:95:04:0a:c4    20 Registered    99-09-03 19:06 99-09-03 19:06 11018   252
00:90:83:32:9f:8c    19 Registered    99-09-03 19:06 99-09-03 19:06 11000   240
00:90:83:36:82:ee    17 Registered    99-09-03 19:06 99-09-03 19:06 10989   208
00:90:83:36:82:f1    17 Registered    99-09-03 19:06 99-09-03 19:06 11000   210
00:90:96:00:29:6d   126 Registered    99-09-03 19:06 99-09-03 19:06 11221   591
00:90:96:00:29:71   126 Registered    99-09-03 19:06 99-09-03 19:06 11216   609
00:90:96:00:39:7f   125 Registered    99-09-03 19:06 99-09-03 19:06 11223   534
00:90:96:00:39:f9   124 Registered    99-09-03 19:06 99-09-03 19:06 11211   636
00:a0:73:69:39:65    15 Registered    99-09-03 19:06 99-09-03 19:06 10984   192

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

The parameters that you can display through the flap list are listed below:

Table 19-9 Flap List Parameters

Parameter	Description
MAC Address	The RF MAC address of the cable modem.
Flap Count	The number of times that this cable modem reset from either the ranging complete or registration complete states.
Last Known State	The last known state of the modem.
Insert Time	The date and time that this cable modem was added to the flap list.

Remove Time	The last date and time that this cable modem reset.
Hit Count	The number of times the modem responds to MAC layer keep alive messages. It can indicate intermittent upstream, laser clipping, or common-path distortion.
Miss Count	Specifies the number of times the cable modem misses the MAC layer keep alive messages. It can indicate intermittent upstream, laser clipping, or common-path distortion.

Clearing the Flap List

To delete all entries in the flap list table on a specific cable interface, perform the following task within interface cable <c/s/i> mode.

Task	Command
Clear the flap list.	**flap-list clear**

Example

```
cli:172.16.19.10:interface:cable:csi(1/1/1)# show flap-list

row count: 13

Mac Address       Flap  Last Known    Insert Time    Remove Time    Hit   Miss
                  Count State                                     Count Count
----------------- ----- ------------- -------------- -------------- ----- -----
00:10:95:01:ef:d8    19 Registered    99-09-03 19:06 99-09-03 19:06 11009   256
00:10:95:01:f0:05    16 Registered    99-09-03 19:06 99-09-03 19:06 10990   208
00:10:95:04:0a:b7    22 Registered    99-09-03 19:06 99-09-03 19:06 11042   304
00:10:95:04:0a:c3    22 Registered    99-09-03 19:06 99-09-03 19:06 11037   304
00:10:95:04:0a:c4    20 Registered    99-09-03 19:06 99-09-03 19:06 11029   252
00:90:83:32:9f:8c    19 Registered    99-09-03 19:06 99-09-03 19:06 11011   240
00:90:83:36:82:ee    17 Registered    99-09-03 19:06 99-09-03 19:06 11000   208
00:90:83:36:82:f1    17 Registered    99-09-03 19:06 99-09-03 19:06 11011   210
00:90:96:00:29:6d   126 Registered    99-09-03 19:06 99-09-03 19:06 11232   591
00:90:96:00:29:71   126 Registered    99-09-03 19:06 99-09-03 19:06 11227   609
00:90:96:00:39:7f   125 Registered    99-09-03 19:06 99-09-03 19:06 11234   534
00:90:96:00:39:f9   124 Registered    99-09-03 19:06 99-09-03 19:06 11222   636
00:a0:73:69:39:65    15 Registered    99-09-03 19:06 99-09-03 19:06 10995   192

cli:172.16.19.10:interface:cable:csi(1/1/1)# flap-list clear
cli:172.16.19.10:interface:cable:csi(1/1/1)# show flap-list

Mac Address       Flap  Last Known    Insert Time    Remove Time    Hit   Miss
                  Count State                                     Count Count
----------------- ----- ------------- -------------- -------------- ----- -----

cli:172.16.19.10:interface:cable:csi(1/1/1)#
```

Managing Quality of Service

Quality of Service (QoS) defines the transmission ordering and scheduling on the Radio Frequency (RF) Interface. To provide for upstream traffic through the cable modem, DOCSIS 1.1 QoS classifies packets traversing the RF MAC interface into a Service Flow. To provide for upstream traffic through the cable modem terminating systems (CMTS), DOCSIS 1.1 QoS classifies packets prior to traversing the RF MAC interface into a Service Flow.The Cuda 12000 and cable modems provide this QoS by shaping, policing, and prioritizing traffic according to a parameter set defined for the Service Flow.

Managing QoS on the Cuda 12000 involves viewing statistics and configuration of cable modems for QoS protocol mechanisms. The protocol mechanisms are the following:

- Service Flows
- Classifiers

The statistical and configuration information for these protocol mechanisms is obtained from the associated configuration and provisioning files that were created for the cable modem during CMTS configuration and cable modem provisioning.

This section describes service flows and classifiers, and the processes for viewing the statistics and configuration associated with these QoS mechanisms. The Cuda 12000 supports QoS as defined by the ***Data-Over-Cable Service Interface Specifications****, RFI SP-RFIv1.1-106001215.*

Service Flows

A Service Flow is a QoS protocol mechanism that serves as a MAC-layer transport service and provides a unidirectional flow of packets transmitted either upstream by the cable modem or downstream by the CMTS. A Service Flow is characterized by a Service Flow ID (SFID), the service ID (SID) and a set of QoS parameters. A SID refers only to packets transmitted upstream.

There are three types of QoS parameter sets, as follows:

Table 19-10 QoS Parameter Set Types

Parameter Set	Description
ProvisionedQosParamSet	Service Flow Provisioned QoS Parameters are in the configuration file and are introduced during registration.
AdmittedQoSParamSet	Service Flow Admitted QoS Parameters define how the CMTS is reserving resources. (The CM may also reserve resources.) Examples of resources that are reserved are bandwidth and other memory or time-based resources that are required to activate the flow.
ActiveQosParamSet	Service Flow Active QoS Parameters define what is actually being provided to the Service Flow.

At registration, at least two Service Flows are defined per cable modem configuration file, one for the upstream and one for the downstream. The first upstream Service Flow describes the primary upstream Service Flow and is the default Service Flow used for upstream unclassified traffic. The first downstream Service Flow describes service to the primary downstream Service Flow and is the default Service Flow used for downstream unclassified traffic.

You define Service Flows when you provision cable modems. For information about provisioning cable modems, refer to the Fast Flow Broadband Provisioning Manager Guide, or the guide for your third-party provisioning vendor.

You can view summaries of DOCSIS 1.1 QoS Parameters for Service Flows. The following tables describe parameters, tasks and commands for viewing Service Flow Summaries, upstream Service Flows and Service Flow Parameter Sets.

Table 19-11 Parameters for Viewing a Summary of all Service Flow

Parameter	Description
Service Flow ID	A 32-bit identifier that the CMTS assigns to a Service Flow. All Service Flows have a SFID.
Direction	Indicates that the Service Flow is a downstream Service flow or an upstream Service flow.
Primary	Indicates whether the Service Flow is the primary or secondary Service Flow.
Time Created	The creation time of the Service Flow.
Class Name	(Optional) The name of the Service Class used by the flow.
Scheduling Type	The scheduling service that the CMTS provides for the Service Flow. The scheduling types are: ■ undefined (appears only on downstream Service Flows) ■ best effort (appears only on upstream Service Flows) ■ non real time polling service (appears only on upstream Service Flows) ■ real time polling service (appears only on upstream Service Flows) ■ unsolicited grant service with AD (appears only on upstream Service Flows) ■ unsolicited grant service (appears only on upstream Service Flows)

Table 19-12 Parameters Contained in Upstream Service Flows Display

Parameter	Description
SID	The Service ID for the upstream Service Flow.
Fragments	The number of fragmentation headers the upstream Service Flow received, regardless if the fragment was correctly re-assembled into a valid packet.
Discarded Fragments	The number of upstream fragments the flow discards and does not assemble into a valid upstream packet.
Concatenated Bursts	The number of concatenation headers the upstream Service Flow received.

Table 19-13 Parameters Contained in Parameter Sets Display

Parameter	Description
Cable Modem	The MAC address of the cable modem for which you are displaying the parameter set.
SFID	The Service Flow ID associated with the displayed parameters.
Param Type	Indicates the parameter set that is being viewed. The parameter types are displayed as follows: ■ 1 indicates Active ■ 2 indicates Admitted ■ 3 indicates Provisioned
Service Class Name	(Optional) Name that identifies the service class used by the Service Flow.
Priority	The relative priority of the Service Flow. A higher value indicates a higher priority.

Parameter	Description
Max Traffic Rate (bits/sec)	Maximum sustained traffic rate, in bits/sec, allowed for this Service Flow. A value of zero indicates no maximum traffic rate is being enforced. This parameter applies to both upstream and downstream Service Flows.
Max Traffic burst (bytes)	Token bucket size, in bytes, for this parameter set. The max traffic burst size and the maximum traffic rate determine the maximum sustained traffic rate.
Min Reserved Rate (bits/sec)	Guaranteed minimum rate, in bits/sec, for this parameter set. The default is zero indicates no reserved bandwidth.
Min Reserved Packet (bytes)	Assumed minimum packet size, in bytes, for the minimum reserved rate.
Active Timeout (secs)	Maximum duration, in seconds, that resources remain unused on an active Service Flow before the CMTS deactivates and de-admits the Service Flow. The default is zero to indicate an infinite amount of time.
Admitted Timeout	Maximum duration, in seconds, that resources remain in admitted state, in excess of active resources, before being released. A value of zero indicates an infinite amount of time.
Max Concat Burst (bytes)	Maximum concatenated burst, in bytes, for an upstream Service Flow. A value of zero indicates no maximum burst.

Parameter	Description
Scheduling Type	Upstream scheduling service for an upstream Service Flow. For downstream Service Flows, the value of this parameter is "undefined."
Nominal Polling Interval (usecs)	Nominal interval, in microseconds, between successive unicast request opportunities on only an upstream Service Flow. This value is zero if this parameter does not apply to the scheduling type of the QoS parameter set or if the value is unknown. Nominal Polling does not apply to downstream flows.
Tolerable Poll Jitter (usecs)	Maximum amount of time, in microseconds, that the unicast request interval delays from the nominal periodic schedule, on only an upstream Service Flow. Tolerable Poll Jitter does not apply to downstream flows.
Unsolicited Grant Size (bytes)	Unsolicited grant size, in bytes. It includes the entire MAC frame data PDU from the Frame Control byte to end of the MAC frame, on only an upstream flow. Unsolicited Grant Size does not apply to downstream flows.
Nominal Grant Interval (usecs)	Nominal interval, in microseconds, between successive data grant opportunities, on only an upstream service flow. Nominal Grant Interval does not apply to downstream flows.
Tolerable Grant Jitter (usecs)	Maximum amount of time, in microseconds, that the transmission opportunities delay from the nominal periodic schedule, on only an upstream flow. Tolerable Grant Jitter does not apply to downstream flows.

Parameter	Description
Grants Per Interval	Number of data grants per nominal grant interval, on only an upstream flow. Grants Per Interval does not apply to downstream flows.
TOS AND Mask	Specifies the AND mask for IP TOS byte overwrite, on only an upstream flow. TOS AND Mask does not apply to downstream flows.
TOS OR Mask	Specifies the OR mask for IP TOS byte overwrite, on only an upstream flow. TOS OR Mask does not apply to downstream flows.
Max Latency (usecs)	Maximum latency between the reception of a packet by the CMTS on its network side interface (NSI) and the forwarding of the packet to the RF interface. A value of zero indicates no maximum latency. This parameter only applies to downstream Service Flows.
Request Policy Octets	Indicates the transmit interval opportunities the cable modem uses for upstream transmission requests. In addition, Request Policy Octets specifies if fragmentation concatenation and PHS are allowed, on only an upstream flow. Request Policy Octets does not apply to downstream flows.
Bit Map	Indicates the set of QoS parameters actually signaled in the DOCSIS registration or dynamic service request message that created the QoS parameter set.

Viewing Service Flows

Perform the following tasks within interface:cable:csi <c/s/i> mode to view Service Flow summaries, upstream Service Flows and Service Flow parameter sets. Views may be defined for all Service Flows, for a specified Service Flow and for a specified cable modem:

Task	Command
1. View a summary of all Service Flows or a specified SFID.	**show modem** [<mac address>] **service-flow** [<sfid>]
2. View a summary of an upstream Service Flow.	**show modem** [<mac address>] **service-flow** [<sfid>] **upstream**
3. View parameter sets for all Service Flows.	**show modem service-flow parameter-set**
4. View Service Flow parameters sets for a specified Service Flow or cable modem.	**show modem** [<mac address>] **service-flow** [<sfid>] **parameter-set**
5. View a specified parameter set. Following are the values that you enter for [param-type]: ■ To view an Active Service Flow parameter set enter 1. ■ To view an Admitted Service Flow parameter set enter 2. ■ To view a Provisioned Service Flow parameter set enter 3.	**show modem** [<mac address>] **service-flow** [<sfid>] **parameter-set** [<param-type>]

Example 1

The following example displays all Service Flows for a specified cable modem:

```
interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:f1 service-flow
Cable Modem:           00:90:83:36:82:f1

row count: 2

Service  Direction  Primary   Time Created   Class Name Scheduling
Flow ID                                               Type
-------- ---------- --------- -------------- ---------- -----------
      21 downstream True      01-07-02 16:03            undefined
      22 upstream   True      01-07-02 16:03            best effort
```

Example 2

The following example displays Upstream Service Flows:

```
interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:f1
service-flow upstream
Cable Modem:           00:90:83:36:82:f1

row count: 1

SFID     Fragments Discarded  Concatenated
                   Fragments  Bursts
-------- --------- ---------- -------------
     256         0          0             0
```

Example 3

The following example displays an Admitted parameter set for a specified Service flow:

```
interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:EE
service-flow 26 parameter-set 2
Cable Modem:          00:90:83:36:82:EE
SFID                                  26
Param Type                       admitted
Service Class Name
Priority                               0
Max Traffic Rate                       0  (bits/sec)
Max Traffic Burst                   1522  (bytes)
Min Reserved Rate                      0  (bits/sec)
Min Reserved Packet                   64  (bytes)
Active Timeout                         0  (secs)
Admitted Timeout                     200  (secs)
Max Concat Burst                       0  (bytes)
Scheduling Type                best effort
Nominal Polling Interval               0  (usecs)
Tolerable Poll Jitter                  0  (usecs)
Unsolicited Grant Size                 0  (bytes)
Nominal Grant Interval                 0  (usecs)
Tolerable Grnat Jitter                 0  (usecs)
Grants Per Interval                    0
TOS AND Mask              11111111
TOS OR Mask               00000000
Max Latency                            0  (usecs)
Request Policy Octets     00:00:00:00
Bit Map
```

NOTE: The following is the bottom half of the parameter set display. It displays the BitMap for the above-mentioned parameters. Off indicates the bit is not set, and on indicates the bit is set.

```
    trafficPriority                   off
    maxTrafficRate                    off
    maxTrafficBurst                   off
    minReservedRate                   off
    minReservedPkt                    off
    activeTimeout                     off
    admittedTimeout                   off
    maxConcatBurst                    off
    schedulingType                    off
    requestPolicy                     off
    nomPollInterval                   off
    tolPollJitter                     off
```

```
unsolicitGrantSize                  off
nomGrantInterval                    off
tolGrantJitter                      off
grantsPerInterval                   off
tosOverwrite                        off
maxLatency                          off
```

Classifiers

This section describes Classifiers and explains the process for viewing Classifiers.

A Classifier is a QoS protocol mechanism that contains a set of matching criteria that applies to each downstream and upstream packet entering the cable network. Downstream Classifiers apply to packets that the CMTS is transmitting. Upstream Classifiers apply to packets that the cable modem is transmitting.

The matching criteria of a Classifier includes a reference to a Service Flow, indicated by a Service Flow ID. For example, incoming packets attempt to match to a Classifier. If the packet matches the Classifier, it is forwarded to the referenced SFID. If the packet does not match a Classifier, it is forwarded to the primary Service Flow. Several Classifiers may refer to the same Service Flow.

You define the matching criteria for Classifiers when you provision cable modems. For information about provisioning cable modems, refer to the Fast Flow Broadband Provisioning Manager CLI-based Administration Guide or the guide for your third-party provisioning vendor.

The following table describes the parameters contained in viewing Service Flow Classifiers:

Table 19-14 Parameters Contained in Viewing Service Flow Classifiers

Parameter	Description
SFID	The Service Flow Identifier.
CID	Unique identifier for the packet classifier that the CMTS assigns.
Direction	Indicates the direction for the classifier.
Priority	Indicates the order of evaluation for the classifiers. The higher the value, the higher the priority. A default value of zero is for provisioned Service Flow classifiers. A default value of 64 is for dynamic Service Flow classifiers.
IP TOS Low	Low value of a range of TOS byte values. If the referenced parameter is not present in the classifier, the value is zero.
IP TOS High	High value of a range of TOS byte values. If the referenced parameter is not present in the classifier, the value is zero.
IP TOS Mask	Mask value that ensures range checking of the TOS Low and TOS High values.

Parameter	Description
IP Protocol	Indicates the value of the IP protocol field necessary for IP packets to match this rule. A value of 256 matches traffic with any IP protocol value. A value of 257 matches both TCP and UDP. If the referenced parameter is not present in the classifier, the value is 258.
IP Src Addr	Indicates the value of the IP source address necessary for packets to match this rule.
IP Src Mask	Specifies the bits of a packet's IP source address to compare when matching this rule.
IP Dest Addr	Specifies the low end inclusive range of TCP/UDP source port numbers to which the packet compares. This parameter is ignored for non-TCP/UDP IP packets. If the referenced parameter is not present in the classifier, the value is zero.
IP Dest Mask	Specifies the bits of a packet's IP destination address to compare when matching this rule.
IP Src Port Start	Specifies the low and inclusive range of TCP/UDP source port numbers to which a packet compares.
IP Src Port End	Specifies the high end inclusive range of TCP/UDP source port numbers to which a packet compares.
IP Dest Port Start	Specifies the low end inclusive range of TCP/UDP destination port number to which a packet compares.

Parameter	Description
IP Dest Port End	Specifies the low end inclusive range of TCP/UDP destination port numbers to which a packet compares.
Dest MAC Addr	Indicates the destination MAC address. An Ethernet packet matches an entry when the destination MAC address equals the destination MAC mask.
Dest MAC Mask	Indicates the destination MAC mask. An Ethernet packet matches an entry when the destination MAC address equals the value of the destination MAC mask.
Src MAC Addr	Indicates the source MAC address. An Ethernet packet matches an entry when the source MAC address equals the value of this parameter.
Enet Protocol Type	Indicates the format of the Layer 3 protocol identifier in the Ethernet packet. The options are:
none	Rule does not use the Layer 3 protocol type as a matching criteria.
ethertype	Rule applies only to frames that contain an Ethertype value.
dsap	Rule applies to frames using IEEE802.3 encapsulation format with a Destination Service Access Point (DSAP).
mac	Rule applies to MAC management messages.
all	Rule applies to all Ethernet packets.

Parameter	Description
Enet Protocol	Indicates the packet class Ethernet protocol. The options are:
none	Parameter is ignored when considering whether a packet matches the current rule.
ethertype	Indicates the 16-bit value of the Ethertype that the packet must match to match the rule.
dsap	Lower 8-bits of the value must match the DSAP byte of the packet to match the rule.
mac	Indicates the lower and upper 8-bits of this object represent the upper and lower bound of MAC management messages.
User Priority Low	Applies to Ethernet frames using the 802.1P/Q tag header. Tagged Ethernet packets must have a 3-bit priority field within the range of the low and high priority to match this rule.
User Priority High	Applies to Ethernet frames using the 802.1P/Qtag header. Tagged Ethernet packets must have a 3-bit priority field within the range of the low and high priority to match this rule.
VLAN Id	Applies to Ethernet frames using the 802.1P/Qtag header. If this parameter is a non-zero value, tagged packets must have a VLAN identifier that matches the value to match the rule.

Parameter	Description
State	Indicates whether or not the classifier is currently classifying packets to a Service Flow. The options are: active or inactive.
Packets	Indicates the number of packets that have been classified.
Bit Map	Indicates what parameter encoding were actually present in the DOCSIS packet classifier encoding in the DOCSIS message that created the classifier.

Viewing Classifiers

Perform the following tasks within interface:cable:csi <c/s/i> mode to view Classifiers. Views may be defined for all Service Flows, for a specified Service Flow and for a specified cable modem:

Task	Command
1. View Classifiers for all Service Flows.	**show modem service-flow classifier**
2. View Service Flow Classifiers for a specific cable modem.	**show modem** [<mac address>] **service-flow classifier** [<cid>]
3. View Service Flow Classifiers for an SFID.	**show modem** [<mac address>] **service-flow** [<sfid>] **classifier** [<cid>]

Service Flow Logs

This section describes Service Flow Logs and the process for viewing and clearing logs.

Service Flow logs contain historical information about Service Flows that are no longer in use.

The following table describes the parameters contained in viewing Service Flow Logs:

Table 19-15 Parameters for Viewing Service Flow Logs

Parameter	Description
Index SFID	The unique index number generated for the log.
CM MAC Address	The MAC address for the cable modem associated with the Service Flow log.
Packets	The number of packets on the specified Service Flow after payload header suppression.
Time Deleted	The time the Service Flow was deleted.
Time Created	The time the Service Flow was created.
Time Active	The total time the Service Flow is active.

Viewing and Clearing Service Flow Logs

Perform the following tasks within interface:cable:csi <c/s/i> mode to view and clear Service Flow logs.

NOTE: Service Flow logs are not indexed by SFID. You may choose to view and clear Service Flow logs for all Service Flows, by cable modem or by the index number of the log.

Task	Command
1. View Service Flow logs for all Service Flows.	**show modem service-flow log**
2. View a Service Flow log by index number.	**show modem service-flow log** [<log-id>]
3. Clear all Service Flow logs.	**clear service-flow log** <all>
4. Clear Service Flow logs for a cable modem.	**clear service-flow log** <mac address>

Example 1

The following example displays Service Flow logs for all Service Flows:

```
show modem service-flow log

row count: 4

Index SFID     CM MAC Address    Packets   Time Deleted   Time Created   Time Active
----- -------- ----------------- --------- -------------- -------------- -----------
 1045       21 ec:5b:20:00:00:ee         6 01-06-30 19:14 01-07-02 16:03    44:49:32
 1046       22 ec:5b:20:00:00:ee       254 01-06-30 19:14 01-07-02 16:03    44:49:32
 1047       23 ec:5b:20:00:00:ee         7 01-06-29 17:09 01-07-01 21:43    52:33:55
 1048       24 ec:5b:20:00:00:ee       228 01-06-29 17:09 01-07-01 21:43    52:33:55
```

Example 2

The following example displays a Service Flow log by index number.

```
show modem service-flow log 1045
Index                                    1045
SFID                                       21
CM MAC Address              ec:5b:20:00:00:ee
Packets                                     6
Time Deleted                   01-06-30 19:14
Time Created                   01-07-02 16:03
Time Active                          44:49:32
```

Dynamic Service

This section describes Dynamic Service and explains the process for viewing Dynamic Service Flow Statistics.

In addition to Service Flow creation at the time the cable modem registers, Dynamic Service creates Service Flows that are defined by the cable modem (CM) or the CMTS.

A CM-initiated DSA-Request contains a Service Flow Reference (SREF) for one upstream and/or one downstream Service Flow, an Admitted and/or Active QoS Parameter set and required Classifiers. A CMTS-initiated DSA-Request contains a SFID for one upstream and/or one downstream Service Flow. It also may contain a SID, an Admitted and/or Active QoS Parameter set and required Classifiers.

The following table describes the parameters contained in viewing Dynamic Service Flow Statistics:

Table 19-16 Parameters for Viewing Dynamic Service Flow Statistics

Parameter	Description
DSA Requests	The number of dynamic service addition requests.
DSA Responses	The number of dynamic service addition responses.
DSA Acks	The number of dynamic service addition acknowledgements.
DSC Requests	The number of dynamic service change requests.
DSC Responses	The number of dynamic service change responses.
DSC Acks	The number of dynamic service change acknowledgements.

Parameter	Description
DSD Requests	The number of dynamic service delete requests.
DSD Responses	The number of dynamic service delete responses.
Dynamic Adds	The number of successful dynamic service addition transactions.
Dynamic Add Fails	The number of failed dynamic service addition transactions.
Dynamic Changes	The number of successful dynamic service change transactions.
Dynamic Change Fails	The number of failed dynamic service change transactions.
Dynamic Deletes	The number of successful dynamic service delete transactions.
Dynamic Delete Fails	The number of failed dynamic service delete transactions.
DCC Requests	The number of dynamic channel change request messages traversing an interface. This value is only non-zero for the downstream.
DCC Responses	The number of dynamic channel change response messages traversing an interface. This value is only non-zero for the upstream.

Parameter	Description
DCCs	The number of successful dynamic channel change transactions. This value is only non-zero for the downstream.
DCC Fails	The number of failed dynamic service change transactions. The value is only non-zero for the downstream.
DCC Acks	The number of dynamic channel change acknowledgment messages traversing an interface. This value is only non-zero for the downstream.

View Dynamic Service Flow Statistics

Perform the following tasks within interface:cable:csi <c/s/i> mode to view Dynamic Service Flow Statistics:

Task	Command
Display Service Flow statistics.	**show dynamic-service stats**

Example

The following example displays Dynamic Service Flow Statistics:

```
cli:show dynamic-service-stats
Direction                              Outbound
DSA Requests                                  0
DSA Responses                                 0
DSA Acks                                      0
DSC Requests                                  0
DSC Responses                                 0
DSC Acks                                      0
DSD Requests                                  0
DSD Responses                                 0
Dynamic Adds                                  0
Dynamic Add Fails                             0
Dynamic Changes                               0
Dynamic Change Fails                          0
Dynamic Deletes                               0
Dynamic Delete Fails                          0
DCC Requests                                  0
DCC Responses                                 0
DCC Acks                                      0
DCCs                                          0
DCC Fails                                     0

Direction                               Inbound
DSA Requests                                  0
DSA Responses                                 0
DSA Acks                                      0
DSC Requests                                  0
DSC Responses                                 0
DSC Acks                                      0
DSD Requests                                  0
DSD Responses                                 0
Dynamic Adds                                  0
Dynamic Add Fails                             0
Dynamic Changes                               0
Dynamic Change Fails                          0
Dynamic Deletes                               0
Dynamic Delete Fails                          0
DCC Requests                                  0
DCC Responses                                 0
DCC Acks                                      0
DCCs                                          0
DCC Fails                                     0
```

20 SUBSCRIBER MANAGEMENT

Through Subscriber Management, the Cuda 12000 provides added security for your cable network against:

- Malicious tampering with the cable modem software
- Unwanted traffic from entering the cable network

To achieve added security, the Cuda 12000 provides protocol filtering to and from the cable modem, and limits the number of IP addresses available to Customer Premise Equipment (CPE) devices.

This chapter contains the following sections:

The Cuda 12000 conforms with the DOCSIS 1.1 IETF Subscriber Management MIB.

About Subscriber Management Filtering

Subscriber Management filtering on the Cuda 12000 works as follows:

1. You configure groups of Subscriber Management IP packet filters. These filter groups provide the source-matching criteria for upstream and downstream traffic for cable modems and CPE devices. These filter groups are used across the Cuda 12000 and are persisted on the Route Server.
2. You assign default filter groups for upstream and downstream traffic for cable modems and CPE devices.
3. During initialization, the cable modem is assigned a filter group that exists on the provisioning server. (*For information about Subscriber Management configuration on the provisioning server, refer to the FastFlow Broadband Provisioning Manager Guide, or the guide for your provisioning manager vendor.)*
4. If Subscriber Management filter groups do not exist on the provisioning server, the cable modem is assigned a default filter group that exists on the Cuda 12000.
5. The Cuda 12000 supports Subscriber Management filter groups on a per cable modem basis. If the network administrator chooses not to use the criteria of the default filter group for a particular cable modem, another filter group may be assigned on the Cuda 12000 to that cable modem. Modifications to default filter groups on a per cable modem basis involve sending SNMP sets directly to the cable modem. The modifications do not persist, so the default filter group is not overwritten.

About CPE Control

In addition to providing added security through filtering, Subscriber Management provides added security by limiting the number of IP addresses available to CPE devices, which minimizes the risk of malicious tampering against your cable network.

Subscriber Management allows a maximum of 16 IP addresses available to CPE devices. Once the limit that you configure is met, packets from additional IP addresses are dropped.

You can:

- Specify a default set of CPE control parameters that apply chassis-wide.
- Configure CPE control parameters on a modem-by-modem basis. When you configure parameters on a modem-by-modem basis, you send SNMP sets directly to the cable modem. These changes do not persist.

Configuring Filter Groups

Configuring filter groups on the Cuda 12000 involves defining the right source matching criteria.

You can configure up to 60 global filter groups. Each filter group may contain up to 40 matching criteria rules (filters).

Before you configure the matching criteria for Subscriber Management filter groups, keep in mind that you may use Subscriber Management filter groups in addition to the IP packet filtering system, which is the default filtering system on the Cuda 12000. The IP packet filtering system and Subscriber Management filters operate in serial fashion. If either the Subscriber Management filters or the IP packet filtering system denies the packet, then the packet is dropped. For example:

- If your network is configured to use both IP packet filtering and Subscriber Management filters, the packet is first filtered through the IP packet filtering matching criteria.
- If IP packet filtering denies the packet, the packet is dropped and is not forwarded.
- If the packet is accepted by IP packet filtering, the packet is filtered through the Subscriber Management filter.
- If the Subscriber Management filter denies the packet, the packet is dropped.
- If the packet is accepted by the Subscriber Management filter, the packet is forwarded.

For detail information about the IP packet filtering system, refer to Chapter 15, IP Packet Filtering beginning on page 327.

The following table lists the matching criteria parameters that you configure to assign global Subscriber Management filter groups on the Cuda 12000:

Table 20-1 Subscriber Management Global Filtering Parameters

Parameter	Description
Group number	Group number specifies the ID of the filter group to which you want the filter to belong. Allowable group number values range from 1 to 60. A value of 0 means that no filtering is performed.
Filter number	Filter number specifies the index number for the filter within the group. Allowable values range from 1 to 40.
Deny or Permit	Specify packet filtering for the particular matching criteria: ■ deny: drop packets that match the filter criteria. ■ permit: forward packets that match the filter criteria.
Source IP Address	The filter attempts to match the source IP address to the IP address in the IP packet. If the addresses match, the filter is applied to the packet.
Source IP Mask	Bit mask that applies to the source IP address prior to matching. The source IP address and mask are the matching criteria for the packet. By default, the source IP address and mask specify a filter that matches all source addresses. This mask entry is not necessarily the same as the subnet mask.
Destination IP Address	The filter attempts to match the destination IP address to the IP address in the IP packet. If the addresses match, the filter is applied to the packet.
Destination IP Mask	Bit mask that applies to the destination IP address prior to matching. The destination IP address and mask are the matching criteria for the packet. By default, the destination IP address and mask specify a filter that matches all destination addresses. The mask entry is not necessarily the same as the subnet mask.

Parameter	Description
Protocol	The protocol that the filter attempts to match. Specify one of the following protocols: TCP, UDP, Any, and Number. ■ You may obtain protocol numbers from the Internet Assigned Numbers Authority (IANA) at www.iana.org. ■ You may specify a protocol number from 0 to 256. Note that specifying 256 is the same as specifying "Any."
TOS Value	The two-digit hexadecimal number indicating the Type of Service (TOS) value to be matched against the TOS value in IP packets (for example, 0a). The default is 00.
TOS Mask	The two-digit hexadecimal number that specifies the mask to be applied to the TOS value matched in the IP packet (for example 1b). The mask determines which bits are matched (a 0 specifies a match while a 1 specifies no match). The default is 00, which means that the TOS value you specify is matched against all TOS values in IP packets.
Source Port (Applies only to TCP or UDP filters.)	Optional. The source TCP or UDP port number to match. Specify one of the following values: ■ Any: Match any source port ■ Number: Match a source port number to the TCP or UDP port. The allowable TCP or UDP port number range is 0 to 65536. Note that specifying 65536 is the same as specifying "Any."
Destination Port (Applies only to TCP or UDP filters.)	Optional. The destination TCP or UDP port number to match. Specify one of the following values: ■ Any: Match any source port ■ Number: Match a source port number to a TCP or UDP port. The allowable TCP or UDP port number range is 0 to 65536. Note that specifying 65536 is the same as specifying "Any."

Parameter	Description
TCP Flag	Optional. The value of the TCP flags. The following is a list of the TCP flag options. Leaving this field blank indicates a null value (no flags). ■ urgent: The TCP segment is marked urgent. ■ ack: The acknowledgement number field in the TCP field segment is significant. ■ push: The TCP software must push all the data sent, so far, through the connection to the receiving application. ■ reset: The connection is reset. ■ syn: The sequence numbers are resynchronized, marking the beginning of a connection. ■ fin: The transmitting CPE has no data to transmit. TCP flags must always be a subset of the TCP flag mask in order for the packet header to be matched. For example, to match all packets where only the "urgent" flag is set, and the mask is set at "syn, and fin," the resulting flag values would be "urgent" and the mask would be: "urgent, syn, fin."
TCP Mask	Optional. The flag of interest in the TCP header for the packet to match. Leaving this field blank indicates a null value (no flags). The following is a list of the options. *Refer to the above "TCP Flag" list for descriptions.* ■ urgent ■ ack ■ push ■ reset ■ syn ■ fin

The source matching criteria that you define for the global filter group is used across the Cuda 12000 and is persisted on the Route Server. To configure matching criteria for global filter groups, perform the following tasks:

Task	Command
1. Enter root mode.	**root**
2. Create a Subscriber Management filter group.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>}
3. Specify the source IP address and mask.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>} [**src** <ip-address> <mask>]
4. Specify the destination IP address and mask.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>} [**dest** <ip-address> <mask>]
5. Specify the TOS value and mask.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>} [**tos** <tos-value> <mask>]
6. Specify the source TCP or UDP port number.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>} [**src-port** {**any |** <number>}]
7. Specify the destination TCP or UDP port number.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>} [**dest-port** {**any |** <number>}]
8. Specify the TCP flag and TCP-flag-mask options.	**cm-filter** <group number> <filter number> {**deny | permit**} **prot** {**any | tcp | udp |** <number>} [**tcp-flag** {**ack | fin | push | reset | syn | urgent**} **tcp-flag-mask** {**ack | fin | push | reset | syn | ask | urgent**}]

Example

The following example displays global Subscriber Management filter group 2 and index 1, which is configured to deny packets, to filter TCP packets and to use a destination IP and mask address of 144.133.1.1 255.255.255.0:

```
cli:192.168.208.3:root# cm-filter 2 1 deny prot tcp dest
144.133.1.1 255.255.255.0
cli:192.168.208.3:root# show cm-filter 2 1
Group                                    2
Index                                    1
Src Address               0.0.0.0
Src Mask                  0.0.0.0
Dest Address              144.133.1.1
Dest Mask                 255.255.255.0
Protocol                                tcp
TOS                       00
TOS Mask                  00
Action                                 deny
Matches                                   0

Source Port                           65536
Destination Port                      65536
TCP Flag Values
TCP Flag Mask
cli:192.168.208.3:root#
```

Viewing Filter Groups

You can view a particular global Subscriber Management filter group or view all filter groups on the cable network. To view Subscriber Management filters, perform the following tasks:

Task	Command
1. View all the Subscriber Management filter groups on the cable network.	**show cm-filter**
2. View a particular Subscriber Management filter.	**show cm-filter** [<group number> [<filter number>]]

Example

The following example displays all the global Subscriber Management filter groups on the cable network. *Refer to the previous section for an example of a display for a particular Subscriber Management filter*:

```
cli:192.168.208.3:root# show cm-filter
row count: 5
Group Index Src Address      Src Mask         Dest Address     Dest Mask
----- ----- ---------------- ---------------- ---------------- ----------------
    1     1 1.1.1.1          255.255.255.0    2.2.2.2          255.255.255.0
    1     2 0.0.0.0          0.0.0.0          0.0.0.0          0.0.0.0
    1     3 0.0.0.0          0.0.0.0          3.3.3.3          255.255.255.0
    1     4 0.0.0.0          0.0.0.0          0.0.0.0          0.0.0.0
    2     1 0.0.0.0          0.0.0.0          0.0.0.0          0.0.0.0
cli:192.168.208.3:root#
```

Deleting Filter Groups and Filters

You can delete a particular Subscriber Management filter group or a filter within the group. To delete Subscriber Management filter groups and filters, perform the following tasks:

Task	Command
1. Delete a Subscriber Management filter group.	**no cm-filter** <group number>
2. Delete a particular Subscriber Management filter within a group.	**no cm-filter** <group number> <filter number>

Modifying Existing Filter Groups

You may replace the source matching criteria for an existing filter group. To do so, perform the same tasks as listed on page 500. For example:

1. Enter root mode.
2. Issue the **show cm-filter** command to identify the filter group you want to modify.
3. Perform Task 2 on page 500.
4. Complete Tasks 3 through 8 on page 500 by specifying the new source matching criteria to replace the existing criteria for the filter group.

Assigning Default Filter Groups

Default filter groups are used by cable modems and CPE devices on all cable interfaces, for upstream and downstream traffic.

You assign four default Subscriber Management filter groups:

- One upstream and one downstream default filter group for cable modems.
- One upstream and one downstream default filter group for CPE devices.

The following applies to default filter groups:

- To assign the default filter group, specify the group ID of a filter group that you create with the **cm-filter** command. This filter group contains the matching criteria to be used as the default filter group.
- Default filter groups are persisted on the Router Server.
- You can use the same default filter group for cable modems and CPE devices.
- You can use the same default filter group for upstream and downstream traffic.

The following table lists the parameters that you set to assign default Subscriber Management filter groups on the Cuda 12000:

Table 20-2 Default Filter Group Parameters

Parameter	Description
CPE DS Filter Group	The ID of the filter group that you specify to be the default filter group, to be used by CPE devices for downstream traffic.
CPE US Filter Group	The ID of the filter group that you specify to be the default filter group, to be used by CPE devices for upstream traffic.
CM DS Filter Group	The ID of the filter group that you specify to be the default filter group, to be used by cable modems for downstream traffic.
CM US Filter Group	The ID of the filter group that you specify to be the default filter group, to be used by cable modems for upstream traffic.

To assign default Subscriber Management filter groups, perform the following tasks:

Task	Command
1. From any mode, assign a filter group to CPE devices for downstream traffic.	**cm-filter-default cpe downstream** <group id>
2. From any mode, assign a filter group to CPE devices for upstream traffic.	**cm-filter-default cpe upstream** <group id>
3. From any mode, assign a filter group to cable modems for downstream traffic.	**cm-filter-default cm downstream** <group id>
4. From any mode, assign a filter group to cable modems for upstream traffic.	**cm-filter-default cm upstream** <group id>

Example

In the following example, the administrator assigns upstream and downstream filter groups for cable modem traffic and CPE device traffic.

```
cli:192.168.208.3:root# show cm-filter

row count: 2

Group Index Src Address     Src Mask        Dest Address    Dest Mask
----- ----- --------------- --------------- --------------- ---------------
   10     1 1.1.1.1         255.255.255.0   2.2.2.2         255.255.255.0
   10     2 201.1.2.0       255.255.255.0   205.1.1.0       255.255.255.0

cli:192.168.208.3:root# cm-filter-default cm downstream 10
cli:192.168.208.3:root# cm-filter-default cm upstream 10
cli:192.168.208.3:root# cm-filter-default cpe downstream 10
cli:192.168.208.3:root# cm-filter-default cpe upstream 10
cli:192.168.208.3:root# show cm-filter-default
CPE DS Filter Group                     10
CPE US Filter Group                     10
CM DS Filter Group                      10
CM US Filter Group                      10
```

Modifying Filter Groups Per Cable Modem

The Cuda 12000 allows the network administrator to temporarily modify the matching criteria of a default filter group on a per cable modem basis.

A default filter group is modified when the network administrator feels it is necessary to use different matching criteria for a particular cable modem or CPE device. The modifications are applied as follows:

- The filter group containing the modifications overrides the matching criteria of the default filter group. These changes do not persist. The next time the cable modem re-initializes, the changes are wiped out and the cable modem uses the default filter group.
- The filter group containing the modifications is a temporary assignment and is not persisted on the Route Server.
- Since a non-default filter group is not persisted, the modifications do not overwrite the initial source matching criteria of the default filter group.

The following table lists the parameters that you set to modify global filter groups on the Cuda 12000:

Table 20-3 Parameters for Modifying Default Filter Groups

Parameter	Description
Group ID	The group ID of the filter group containing the modifications that will override the default filter group.
IP Address	The IP address of: ■ the cable modem, to which the filter group containing the modifications is applied; or ■ the cable modem that provides network access to the CPE, to which the filter group containing the modifications is applied.
MAC Address	The MAC address of: ■ the cable modem, to which the filter group containing the modifications is applied; or ■ the cable modem that provides network access to the CPE, to which the filter group containing the modifications is applied.

Parameter	Description
SID	A session identifier that detects whether the packet refers to a cable modem or CPE device. ■ For upstream packets, the SID is identified in the "parm2" field of the packet structure. ■ For downstream packets, an ARP lookup is performed to determine for which cable modem the packet is destined.

To modify default Subscriber Management filter groups, perform the following tasks.

- Use the **show modem** command to determine the IP address, MAC address and service ID (SID):
- Use the **show cm-filter** command to determine the group id.

Task	Command
1. Enter interface mode.	**interface:cable:csi** <c/s/i>
2. Modify a default filter group for a CPE device for downstream traffic.	**cm modify cpe-downstream** <group id> {<ip address> | <mac address> | <sid>}
3. Modify a default filter group for a CPE device for upstream traffic.	**cm modify cpe-upstream** <group id> {<ip address> | <mac address> | <sid>}
4. Modify a default filter group for a cable modem for downstream traffic.	**cm modify cm-downstream** <group id> {<ip address> | <mac address> | <sid>}
5. Modify a global filter group for a cable modem for upstream traffic.	**cm modify cm-upstream** <group id> {<ip address> | <mac address> | <sid>}

Example

In this example, the administrator enters interface mode for a cable interface. Then, the administrator assigns a filter group with a group number of 10 to filter upstream and downstream traffic for a cable modem with an SID of 12.

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:96:00:29:71 201.1.1.102        1    1   0 1:2      0   2218 Registered
00:90:96:00:29:6d 201.1.1.103        2    1   0 1:2      0   2215 Registered
00:10:95:01:f0:05 201.1.1.100        3    1   0 1:2      0   2209 Registered
00:10:95:04:0a:c3 0.0.0.0            4    0   0 1:2     -9   2723 Ranging
00:10:95:04:0a:b7 201.1.1.101        5    1   0 1:2      0   2724 Registered
00:90:96:00:39:7f 201.1.1.107        6    1   0 1:2      0   2219 Registered
00:90:96:00:39:f9 201.1.1.105        7    1   0 1:2      0   2216 Registered
00:10:95:04:0a:c4 201.1.1.104        8    1   0 1:2      0   2729 Registered
00:10:95:01:ef:d8 201.1.1.106        9    1   0 1:2      0 524732 Registered
00:90:83:36:82:f1 201.1.1.108       10    1   0 1:2      0   1258 Registered
00:90:83:36:82:ee 201.1.1.109       11    1   0 1:2      0   1249 Registered
00:90:83:32:9f:8c 201.1.1.110       12    1   0 1:2      0   1672 Registered

cli:192.168.208.3:interface:cable:csi(1/1/1)# show cm-filter

row count: 2

Group Index Src Address     Src Mask        Dest Address    Dest Mask
----- ----- --------------- --------------- --------------- ---------------
   10     1 1.1.1.1         255.255.255.0   2.2.2.2         255.255.255.0
   10     2 201.1.2.0       255.255.255.0   205.1.1.0       255.255.255.0

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify cm-downstream 10 12
cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify cm-upstream 10 12
```

Viewing Filter Group Assignments

You can view the Subscriber Management filter groups currently assigned to a particular cable modem and associated CPE devices. To view filter group assignments, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface:cable:csi** <c/s/i>
2. Display filter group assignments for all cable modems (and associated CPE devices).	**show modem cm-filter**
3. Display filter group assignments for a specific cable modem (and associated CPE devices).	**show modem** <mac-address> **cm-filter**

Example

The following example displays filter group assignments for:

- All cable modems and associated CPE devices
- A specific cable modem and associated CPE devices

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem cm-filter

row count: 12

MAC Address       IP Address      CPE DS     CPE US     CM DS      CM US
                                  Filter     Filter     Filter     Filter
                                  Group      Group      Group      Group
----------------- --------------- ---------- ---------- ---------- ----------
00:90:96:00:29:71 201.1.1.102     0          10         10         0
00:90:96:00:29:6d 201.1.1.103     0          10         10         0
00:10:95:01:f0:05 201.1.1.100     10         10         10         10
00:10:95:04:0a:c3 0.0.0.0         0          0          0          0
00:10:95:04:0a:b7 201.1.1.101     0          10         10         0
00:90:96:00:39:7f 201.1.1.107     0          10         10         0
00:90:96:00:39:f9 201.1.1.105     0          10         15         0
00:10:95:04:0a:c4 201.1.1.104     0          10         10         0
00:10:95:01:ef:d8 201.1.1.106     0          10         10         0
00:90:83:36:82:f1 201.1.1.108     0          10         10         0
00:90:83:36:82:ee 201.1.1.109     0          10         10         0
00:90:83:32:9f:8c 201.1.1.110     0          10         10         10

cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem 00:90:96:00:29:71
cm-filter
S l o t                                 1
MAC Address                00:90:96:00:29:71
IP Address                       201.1.1.102

CPE DS Filter Group                      0
CPE US Filter Group                     10
CM DS Filter Group                      10
CM US Filter Group                       0
```

Configuring CPE Control Parameters

In addition to providing added security through filtering, Subscriber Management provides added security by limiting the number of IP addresses available to CPE devices, which minimizes the risk of malicious tampering against your cable network.

Subscriber Management allows a maximum of 16 IP addresses available to CPE devices. Once the limit that you configure is met, packets from additional IP addresses are dropped.

The following table lists the parameters that you set to configure Subscriber Management control of CPE devices:

Table 20-4 Subscriber Management Control Parameters

Parameter	Description
Active	Indicates whether Subscriber Management (filtering, enforcement of address limits, etc.) is to be used at the CPE control level. By default, Subscriber Management is disabled.
Max IP	Number of simultaneous IP addresses you can attach to the CPE device. If this parameter is set to zero, the cable modem drops all CPE traffic. The allowable range is 0 to 16.
Learnable	Specifies whether the CMTS learns all CPE IP addresses on the current interface. By default, the CMTS learns all CPE IP addresses on current interfaces.

You can set chassis-wide defaults for all of these parameters or you can set these parameters on a modem-by-modem basis. Refer to "Modifying CPE Control Parameters Per Cable Modem" on page 515 for more information on setting these parameters modem-by-modem.

To set chassis-wide Subscriber Management defaults for CPE devices, perform the following tasks:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. From any mode, specify that you do not want to disable Subscriber Management on all cable interfaces on the chassis.	**no cpe-control active**
3. From any mode, specify that you want to enable Subscriber Management on all cable interfaces on the chassis.	**cpe-control active**
4. From any mode, specify the maximum number of IP addresses available to CPE devices on all cable interfaces on the chassis.	**cpe-control max-ip** <0..16>
5. From any mode, remove the maximum number of IP addresses available to CPE devices on all cable interfaces on the chassis.	**no cpe-control max-ip**
6. From any mode, specify that you want the CMTS to learn all IP addresses of CPE devices on all interfaces on the chassis.	**cpe-control learnable**
7. From any mode, specify that you do not want the CMTS to learn all IP addresses of CPE devices on all interfaces on the chassis.	**no cpe-control learnable**
8. From any mode, display the chassis-wide defaults for Subscriber Management CPE control.	**show cpe-control**

Example

In this example, Subscriber Management of CPE devices is enabled, the maximum number of IP addresses available to CPE devices per cable modem is set to 10, and the ability of the CMTS to learn CPE IP addresses is enabled.

```
cli:192.168.208.3:root# cpe-control active
cli:192.168.208.3:root# cpe-control max-ip 10
cli:192.168.208.3:root# cpe-control learnable
cli:192.168.208.3:root# show cpe-control
MAX IP                                  10
Active                                True
Learnable                             True
```

Modifying CPE Control Parameters Per Cable Modem

The Cuda 12000 allows the network administrator to modify Subscriber Management CPE control per cable modem. Modifications on a per-cable-modem basis are not persisted on the Route Server, and are sent in the form of SNMP sets directly to the cable modem. The next time the cable modem re-initializes, the changes are wiped out and the cable modem uses the default CPE control settings.

Per cable modem, you can:

- Specify whether or not you want to apply Subscriber Management to CPE devices associated with the cable modem.
- Change the maximum number of IP addresses available to the CPE devices associated with the cable modem.
- Specify you want IP addresses of CPE devices associated with the cable modem to be learned by the CMTS.

The following table lists the parameters that you set to modify Subscriber Management CPE control:

Table 20-5 Parameters for Modifying Subscriber Management CPE Control

Parameter	Description
IP Address	The IP address of the cable modem that provides network access to the CPE devices you want to control.
MAC Address	The MAC address of the cable modem that provides network access to the CPE devices you want to control.
SID	A service identifier of the cable modem that provides network access to the CPE devices you want to control.

To modify Subscriber Management CPE control per cable modem, perform the following tasks. Note that you use the **show modem** command to determine the IP address, MAC address and service ID of the cable modem:

Task	Command
1. Enter cable interface mode.	**interface cable** <c/s/i>
2. Specify that you want to use Subscriber Management for CPE devices that use the cable modem to access the network.	**cm modify active** {<ip address> | <mac address> | <sid>}
3. Specify that you do not want to use Subscriber Management for CPE devices that use the cable modem to access the network.	**no cm modify active** {<ip address> | <mac address> | <sid>}
4. Change the maximum number of IP addresses available to CPE devices that use the cable modem to access the network.	**cm modify max-ip** <0..16> {<ip address> | <mac address> | <sid>}
5. Specify that you want the CMTS to learn all IP addresses for CPE devices that use the cable modem to access the network.	**cm modify learnable** {<ip address> | <mac address> | <sid>}
6. Specify that you do not want the CMTS to learn all IP addresses for CPE devices that use the cable modem to access the network.	**no cm modify learnable** {<ip address> | <mac address> | <sid>}
7. Display the Subscriber Management CPE control configuration, per cable modem.	**show modem cpe-control**

Example

In this example, the administrator restricts CPE devices that use a specific cable modem (201.1.1.101) to access the network to five IP addresses.

```
cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem

row count: 12

MAC Address       IP Address      SID  CID  CPE D:U Power  Timing Modem
                                                   (dbMV)        State
----------------- --------------- ---- ---- --- --- ------ ------ ----------
00:90:96:00:29:71 201.1.1.102       1    1   0 1:2      0   2218 Registered
00:90:96:00:29:6d 201.1.1.103       2    1   0 1:2      0   2216 Registered
00:10:95:01:f0:05 201.1.1.100       3    1   0 1:2      0   2208 Registered
00:10:95:04:0a:c3 0.0.0.0           4    0   0 1:2     -9   2723 Ranging
00:10:95:04:0a:b7 201.1.1.101       5    1   0 1:2      0   2725 Registered
00:90:96:00:39:7f 201.1.1.107       6    1   0 1:2      0   2218 Registered
00:90:96:00:39:f9 201.1.1.105       7    1   0 1:2      0   2216 Registered
00:10:95:04:0a:c4 201.1.1.104       8    1   0 1:2      0   2728 Registered
00:10:95:01:ef:d8 201.1.1.106       9    1   0 1:2      0 530199 Registered
00:90:83:36:82:f1 201.1.1.108      10    1   0 1:2      0   1258 Registered
00:90:83:36:82:ee 201.1.1.109      11    1   0 1:2      0   1250 Registered
00:90:83:32:9f:8c 201.1.1.110      12    1   0 1:2      0   1671 Registered

cli:192.168.208.3:interface:cable:csi(1/1/1)# cm modify max-ip 5 201.1.1.101
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem cpe-control

row count: 12

MAC Address       IP Address      MAX IP     Active     Learnable
----------------- --------------- ---------- ---------- ----------
00:90:96:00:29:71 201.1.1.102     16         True       True
00:90:96:00:29:6d 201.1.1.103     16         True       True
00:10:95:01:f0:05 201.1.1.100     16         True       True
00:10:95:04:0a:c3 0.0.0.0         16         True       True
00:10:95:04:0a:b7 201.1.1.101     5          True       True
00:90:96:00:39:7f 201.1.1.107     16         True       True
00:90:96:00:39:f9 201.1.1.105     16         True       True
00:10:95:04:0a:c4 201.1.1.104     16         True       True
00:10:95:01:ef:d8 201.1.1.106     16         True       True
00:90:83:36:82:f1 201.1.1.108     16         True       True
00:90:83:36:82:ee 201.1.1.109     16         True       True
00:90:83:32:9f:8c 201.1.1.110     16         True       True
```

Viewing CPE Control Parameters and CPE Devices

You can view CPE control parameters and CPE devices.

Viewing CPE Control Parameters

To view CPE control parameters, perform the following tasks:

Task	Command
1. From any mode, view default CPE control parameters.	**show cpe-control**
2. From interface:cable:csi<c/s/i> mode, view CPE control parameters for all cable modems using the interface.	**show modem cpe-control**
3. From interface:cable:csi<c/s/i> mode, view CPE control parameters for a specific cable modem using the interface (specify the cable modem's MAC address).	**show modem** <mac-address> **cpe-control**

Example

In this example, the administrator displays default CPE control parameters and then displays CPE control parameters for all cable modems on a specific interface.

```
cli:192.168.208.3:root# show cpe-control
MAX IP                                    16
Active                                  True
Learnable                               True

cli:192.168.208.3:root# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem cpe-control

row count: 12

MAC Address       IP Address      MAX IP     Active     Learnable
----------------- --------------- ---------- ---------- ----------
00:90:96:00:29:71 201.1.1.102     16         True       True
00:90:96:00:29:6d 201.1.1.103     16         True       True
00:10:95:01:f0:05 201.1.1.100     16         True       True
00:10:95:04:0a:c3 0.0.0.0         16         True       True
00:10:95:04:0a:b7 201.1.1.101     5          True       True
00:90:96:00:39:7f 201.1.1.107     16         True       True
00:90:96:00:39:f9 201.1.1.105     16         True       True
00:10:95:04:0a:c4 201.1.1.104     16         True       True
00:10:95:01:ef:d8 201.1.1.106     10         True       True
00:90:83:36:82:f1 201.1.1.108     16         True       True
00:90:83:36:82:ee 201.1.1.109     16         True       True
00:90:83:32:9f:8c 201.1.1.110     16         True       True
```

Viewing CPE Devices

You can view CPE devices associated with all cable modems on an interface or a specific cable modem. To view CPE devices, perform the following tasks:

To view CPE devices, perform the following tasks:

Task	Command
1. Enter interface cable mode.	**interface** <c/s/i>
2. View CPE devices associated with all cable modems.	**show modem cpe-hosts**
3. View CPE devices associated with a specific cable modem.	**show modem** <mac-address> **cpe-hosts**

Example

In this example, the administrator displays the CPE devices associated with a specific cable modem.

```
cli:interface:cable:csi(1/1/1)# show modem 00:90:83:36:82:f1
cpe-hosts
MAC Address                 00:90:83:36:82:f1
IP Address                        201.1.1.111

row count: 1

CPE IP Address   CPE MAC Address   Source
---------------- ----------------- -----------
201.1.2.100      00:b0:d0:72:b2:93     Learned
```

21 MIB BROWSING

The Cuda 12000 supports MIB browsing of cable modems and embedded Multimedia Terminal Adapters (MTAs). This chapter provides information on how to browse cable modem and MTA MIBs, and the MIB objects that are returned.

The cable modem and MTA MIB tables are in compliance with *DOCSIS Operations Support System Interface Specification SP-OSSIv1.1-103-001220; DOCSIS Baseline Privacy Plus Interface Specification SP-BPI+-106001215*; *PacketCable Security Specification PKT-SP-SEC_I02-001229*; RFC2669, RFC2670, and RFC3083.

Cable Modem MIBs

The following is a list and description of the cable modem MIB tables that are supported by the Cuda 12000:

Table 21-1 Cable Modem MIB Tables

MIB Table	Description
docsIfCmMacTable	Describes the attributes of each cable modem MAC interface.
docsIfCmServiceTable	Describes the attributes of each upstream service queue.
docsIfCmStatusTable	Maintains status objects and counters for cable modems.
docsIfDownstreamChannelTable	Describes the attributes of the downstream channel.
docsIfUpstreamChannelTable	Describes the attributes of attached upstream channels. This table is implemented on both the CMTS and cable modem.
docsIfSignalQualityTable	Describes PHY signal quality for downstream channels.
docsIfQosProfileTable	Describes the attributes for each class of service.
docsBpiCmBaseTable	Describes the basic- and authorization-related Baseline Privacy attributes of each cable modem MAC interface
docsBpiCmTEKTable	Describes the attributes of each CM Traffic Encryption Key (TEK) association. The CM maintains (no more than) one TEK association per SID per CM MAC interface.
docsBpi2CmBaseTable	Describes the basic- and authorization-related Baseline Privacy Plus attributes of each cable modem MAC interface.
docsBpi2CmTEKTable	Describes the attributes of each CM Traffic Encryption Key (TEK) association for BPI Plus.
systemGroup	Provides system identification, such as, contact name, device name and location.

MIB Table	Description
subset of ifTable & ifXTable	Provides status information and statistics on interface activity.
docsDevBaseGroup, docsDevSoftwareGroup, docsDevServerGroup	Provides objects needed for cable device system management.
docsDevEvControl	Provides control and logging for event reporting, and contains the following MIB tables: ■ docsDevEvSyslog ■ docsDevEvThrottleAdminStatus ■ docsDevEvThrottleInhibited ■ docsDevEvThrottleThreshold ■ docsDevEvThrottleInterval

MTA MIBs

The following is a list and description of the MTA MIB tables that are supported by the Cuda 12000:

Table 21-2 MTA MIB Tables

MIB Table	Description
pktcMtaDevBase	Provide general information regarding the MTA device for the particular interface.
pktcMtaDevServer	Provides the information that the MTA device uses to initialize when it boots up.
pktcMtaDevSecurity	Provides the public key certificates and other security-related information for the MTA device.
pktcSigDevConfigObjects, pktcSigDevCodecTable, pktcSigEndPntConfigTable	Contains information regarding Display Network Call Signaling (NCS). NCS displays include values for the following parameters. ■ Service-level Configuration ■ CODEC Conversion Types ■ End Point IDs

Browsing Cable Modem and MTA Status

The Cuda 12000 supports the retrieval and display of status information that is maintained by individual cable modems and MTAs connected to the HFC network. This information is useful when you have to monitor the network and troubleshoot network problems.

To retrieve and display this status information:

1. Issue the **interface** <c/s/i> command to access interface mode for the CMTS interface used by the cable modems and MTAs of interest.
2. Issue **show modem** commands (with the appropriate arguments) to retrieve this information, as shown in the following table:

Table 21-3 Commands for Accessing Cable Modem and MTA MIB Tables

Command	MIB Table or Group
show modem <mac address> **cm mac**	CM MAC (docsIfCmMacTable). Refer to Table 21-4 on page 528.
show modem <mac address> **cm service**	CM Service (docsIfCMServiceTable). Refer to Table 21-5 on page 528.
show modem <mac address> **cm status**	CM Status (docsIfCMStatusTable). Refer to Table 21-6 on page 529.
show modem <mac address> **cm downstream**	Downstream (docsIfDownstreamChannelTable). Refer to Table 21-7 on page 530.
show modem <mac address> **cm upstream**	Upstream (docsIfUpstreamChannelTable). Refer to Table 21-8 on page 531.
show modem <mac address> **cm signal-quality**	Signal Quality (docsIfSignalQualityTable). Refer to Table 21-9 on page 532.
show modem <mac address> **cm qosprofile**	QOS (docsIfQosProfileTable). Refer to Table 21-10 on page 533.
show modem <mac address> **cm bpi-base**	BPI Base (docsBPICMBaseTable). Refer to Table 21-11 on page 534.
show modem <mac address> **cm bpi-tek**	BPI TEK (docsBPICMTEKTable). Refer to Table 21-12 on page 538.

Command	MIB Table or Group
show modem <mac address> **cm bpi-plus base**	BPI Plus Base (docsBpi2CmBaseTable). Refer to Table 21-13 on page 540.
show modem <mac address> **cm bpi-plus tek**	BPI Plus TEK (docsBPI2CmTEKTable). Refer to Table 21-14 on page 545.
show modem <mac address> **cm system**	System (systemGroupTable). Refer to Table 21-15 on page 547.
show modem <mac address> **cm device**	Device (docsDevBase, docsDevSoftware, docsDevServer). Refer to Table 21-16 on page 547.
show modem <mac address> **cm device event config**	Device Event Configuration (docsDevEvControlTable). Refer to Table 21-17 on page 550.
show modem <mac address> **cm device event list**	Device Event List (docsDevEventTable). Refer to Table 21-18 on page 551.
show modem <mac address> **cm device event control**	Device Event Control (docsDevEvControlTable). Refer to Table 21-19 on page 552.
show modem <mac address> **cm interface**	Interface (ifTable and ifXTable). Refer to Table 21-20 on page 553.
show mta <mac-address> **base**	MTA Base (pktcMtaDevBaseTable). Refer to Table 21-21 on page 555.
show mta <mac-address> **server**	MTA Server (pktcMtaDevServerTable). Refer to Table 21-22 on page 556.
show mta <mac-address> **security**	MTA Security (pktcMtaDevSecurityTable). Refer to .
show mta <mac-address> **ncs config**	MTA Signalling Configuration (pktcSigDevConfigObjects). Refer to Table 21-23 on page 557.
show mta <mac-address> **ncs codec**	MTA Codec (pktcSigDevCodecTable). Refer to Table 21-24 on page 557.
show mta <mac-address> **ncs endpoint**	MTA Endpoint (pktcSigEndPntConfigTable). Refer to Table 21-25 on page 558.

For example, to display cable modem device status, you would issue the following command:

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# show modem
00:90:83:57:52:5b cm downstream
S l o t                                   1
MAC Address                 00:90:83:57:52:5b
IP Address                        172.30.1.13

docsIfDownChannelId                       1
docsIfDownChannelFrequency        507000000
docsIfDownChannelWidth              6000000
docsIfDownChannelModulatio                3
docsIfDownChannelInterleav                5
docsIfDownChannelPower                    3
docsIfDownChannelAnnex                    4
```

The corresponding output from the cable modem's MIB would be:

```
snmptalk@172.30.1.13> walk docsifdownstreamchanneltable

docsIfDownChannelId.3             Number: 1
docsIfDownChannelFrequency.3      Number: 507000000
docsIfDownChannelWidth.3          Number: 6000000
docsIfDownChannelModulation.3     Number: qam64 (3)
docsIfDownChannelInterleave.3     Number: taps32Increment4 (5)
docsIfDownChannelPower.3          Number: 3
docsIfDownChannelAnnex.3          Number: annexB (4)
```

Cable Modem and MTA Command Output Descriptions

Table 21-4 docsIfCmMacTable Parameters

CLI Output	Description
docsIfCmCmtsAddress	MAC address of the CMTS that is believed to control this MAC domain. At the cable modem, this the source address from the SYNC, MAP, and other MAC-layer messages. If the CMTS is unknown, this value is 00-00-00-00-00-00.
docsIfCmCapabilites	Capabilities of the MAC implementation at this interface.
docsIfCmRangingRespTimeout	Waiting time for a ranging response packet.
docsIfCmRangngTimeout	Waiting time for a ranging timeout packet.

Table 21-5 docsIfCMServiceTable Parameters

CLI Output	Description
docsIfCmServiceQoSProfile	Indicates the QoS attributes that associate with this particular service. A value of zero indicates no associated profile.
docsIfCmServiceTXSlotsImme	Number of upstream mini-slots that were used to transmit data PDUs in immediate contention mode. This includes only PDUs that are presumed to have arrived at the headend. It does not include re-transmission attempts or mini-slots used by Requests.
docsIfCmServiceTx SlotsDed	Number of upstream mini-slots that are used to transmit data PDUs in dedicated mode. For example, as a result of a unicast data grant.
dosIfCmServiceTXRetries	Number of attempts to transmit data PDUs containing requests for acknowledgement that did not result in acknowledgement.
docsIfCmServiceTxExceededs	Number of data PDUs transmission failures due to excessive retries without acknowledgement.
docsIfCmServiceRqRetries	Number of attempts to transmit bandwidth requests that did not result in acknowledgement.

CLI Output	Description
docsIfCmServiceRqExceededs	Number of requests for bandwidth that failed due to excessive retries without acknowledgement.

Table 21-6 docsIfCmStatusTable Parameters

CLI Output	Description
docsIfCmServiceStatusValue	Current cable modem connectivity state.
docsIfCmServiceStatusCode	Status code for this cable modem. This value consists of a single character indicating an error group and two or three numbers indicating the status condition.
docsIfCmServiceTxPower	Operational transmit power for the attached upstream channel.
docsIfCmServiceResets	Number of times the cable modem resets or initializes.
docsIfCmServiceLostSyncs	Number of times the cable modem lost synchronization with the downstream channel.
docsIfCmServiceInvalidMaps	Number of times the cable modem receives invalid MAP messages.
docsIfCmServiceInvalidUcds	Number of times the cable modem receives invalid UCD messages.
docsIfCmServiceInvalidRangi	Number of times the cable modem receives invalid ranging response messages.
docsIfCmServiceInvalidRegis	Number of times the cable modem receives invalid registration response messages.
docsIfCmServiceT1Timeouts	Number of times counter T1 expires in the cable modem.
docsIfCmServiceT2Timeouts	Number of times counter T2 expires in the cable modem.
docsIfCmServiceT3Timeouts	Number of times counter T3 expires in the cable modem.
docsIfCmServiceT4Timeouts	Number of times counter T4 expires in the cable modem.
docsIfCmServiceRangingAbort	Number of times the ranging process was aborted by the CMTS.

Table 21-7 docsIfDownstreamChannelTable

CLI Output	Description
docsIfDownChannelID	CMTS identification of the downstream channel within this particular MAC interface. If the interface is down, the most current value displays. If the channel ID is unknown, a value of zero displays.
docsIfDownChannelFrequency	Center of the downstream frequency, in hertz, associated with this channel. This object returns the current tuner frequency. If a CMTS provides IF output, a value of zero displays unless the CMTS is in control of the final downstream RF frequency.
docsIfDownChannelWidth	Bandwidth, in hertz, of this downstream channel.
docsIfDownChannelModulation	Modulation type for this downstream channel. If the interface is down, this object either returns the configured value from the CMTS, the most current value from the cable modem, or a value of unknown. The options are: unknown, other, qam64, or qam256.
docsIfDownChannelInterleave	Forward Error Correction (FEC) interleaving for this downstream channel.
docsIfDownChannelPower	At the CMTS, the operational transmit power. At the cable modem, the received power level. You can set this parameter to zero at the cable modem is power level management is not supported. If the interface is down, the value is either the value configured at the CMTS, the most current value from the cable modem, or a value of zero.
docsIfDownChannelAnnex	*MIB browsing for this field is not supported in this release.*

Table 21-8 docsIfUpstreamChannelTable Parameters

CLI Output	Description
docsIfUpChannelId	CMTS identification of the upstream channel within this particular MAC interface. If the interface is down, the most current value displays. If the channel ID is unknown, a value of zero displays.
docsIfUpChannelFrequency	Center of the downstream frequency, in hertz, associated with this channel. This object returns a value of zero if the frequency is undefined or unknown,
docsIfUpChannelWidth	Bandwidth, in hertz, of this upstream channel.
docsIfDownChannelModulation Profile	Modulation profile for this upstream channel.
docsIfUpChannelSlotSize	Number of 6.25 microsecond ticks in each upstream mini-slot.
docsIfUpChannelTxTimingOffset	Timing, in units of 6.25 microseconds, of cable modem upstream transmissions to ensure synchronized arrivals at the CMTS. The value indicates the current round trip time at the cable modem or the maximum round trip time seen at the CMTS.
docsIfUpChannelRangingBackoff Start	Initial random backoff window to use when retrying ranging requests. This is expressed as power of 2. For example, a value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used.
docsIfUpChannelRangingBackoff End	Final random backoff window to use when retrying ranging requests. Expressed as a power of 2. For example, a value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used.
docsIfUpChannelTxBackoffStart	Initial random backoff window to use when retrying transmissions. This is expressed as power of 2. For example, a value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used.

CLI Output	Description
docsIfUpChannelTxBackoffEnd	Final random backoff window to use when retrying transmissions. Expressed as a power of 2. For example, a value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used.

Table 21-9 docsIfSignalQualityTable

CLI Output	Description
docsIfSigQIncludes Contention	Indicates the signal includes contention. The options are:
True	CMTS includes contention intervals. A value of 1 indicates True.
False	CMTS does not include contention intervals. This parameter is always False for cable modems. A value of 2 indicates False.
docsIfSigQUnerroreds	Number of codewords this channel receives without error.
docsIfSigQCorrectededs	Number of codewords that this channel receives with correctable errors.
docsIfSigQUncorrectables	Number of codewords this channel received with uncorrectable errors.
docsIfSigQSignalNoise	Signal/Noise ratio, in dB, for this channel. At the cable modem, this is the signal/noise ration of this downstream channel. At the CMTS, this is the average signal/noise of the upstream channel.
docsIfSigQMicroreflections	Total number of microreflections on this interface.
docsIfSigQEqualization Data	At the cable modem, this value indicates the equalization data for the downstream channel. At the CMTS, this value indicates the average equalization data for the upstream channel.

Table 21-10 docsIfQosProfileTable Parameters

CLI Output	Description
docsIfQosProfPriority	Relative priority assigned to this service when allocating bandwidth. Zero indicates lowest priority, and seven indicates the highest priority.
docsIfQosProfMaxUpBandwidth	Maximum upstream bandwidth, in bps, the service allows with this service class.
docsIfQosProfGuarUpBandwidth	Minimum guaranteed upstream bandwidth, in bps, the service allows with this service class.
docsIfQosProfMaxDownBandwidth	Maximum downstream bandwidth, in bps, the service allows with this service class.
docsIfQosProfMaxTxBurst	Maximum number of mini-slots that may be requested for a single upstream transmission. A value of zero indicates no limit.
docsIfQosProfBaselinePrivacy	Indicates whether Baseline Privacy is enabled for this service class.
docsIfQosProfStatus	Creates or deletes rows in the table. You must not change a row while it is active.

Table 21-11 docsBpiCmBaseTable Parameters

CLI Output		Description
docsBpiCmPrivacyEnable		Indicates if the cable modem is provisioned to run Baseline Privacy.
docsBpiCmPublicKey		Indicates the DER-encoded RSA public key corresponding to the public key of the cable modem.
docsBpiCmAuthState The options are:		State of the cable modem authorization Finite State Machine (FSM).
	Start	FSM is in its initial state.
	authwait	The cable modem has received the "Provisioned" event, indicating that it has completed RF MAC registration with the CMTS. In response to receiving the event, the cable modem has sent both an Authentication information and an Authorize Request message to the CMTS and is waiting for the reply.
	Authorized	The cable modem has received an Authorization Reply message which contains a list of valid SAIDs for this cable modem. At this point, the modem has a valid Authorization Key and SAID list. Transition into this state triggers the creation of on TEK FSM for each of the cable modem's privacy-enabled SAIDs.
	ReauthWait	The cable modem has an outstanding re-authorization request. The cable modem was either about to time out its current authorization or received an indication (an Authorization Invalid message from the CMTS) that its authorization was no longer valid. The cable modem sent an Authorization Request message to the CMTS and is waiting for a response.

CLI Output	Description
Auth Reject Wait	The cable modem received an Authorization Reject message in response to its last Authorization Request. The Authorization Reject's error code indicated that the error was not of a permanent nature. In response to receiving this reject message, the cable modem set a timer and transitioned to the Authorized Reject Wait state. The cable modem remains in this state until the timer expires.
docsBpiCmAuthKeySequence Number	Authorized key sequence number of this FSM.
docsBpiCmAuthExpires	Actual clock time when the current authorization for this FSM expires. If the cable modem does not have an active authorization, the value is the expiration date and time of the last active authorization.
docsBpiCmAuthReset	Determines the reauthorize event status.
The options are:	
True	Generates a reauthorize event in the authorization FSM
False	Does not generate an authorization event.
docsBpiCmAuthGraceTime	Grace time for an authorization key. A cable modem is expected to start trying for a new authorization key beginning the grace time number of seconds before the authorization key actually expires. This value cannot change while the authorization state machine is operating.
docsBpiCmTEKGraceTime	The TEK Grace Time in seconds before TEK expires.
docsBpiCmAuthWaitTimeout	The authorize wait timeout. This value cannot change while the authorization state machine is operating.
docsBpiCmReauthWaitTimeout	Reauthorize wait timeout, in seconds. This value cannot change while the authorization state machine is operating.

CLI Output		Description
docsBpiCmOpWaitTimeout		Operational wait timeout, in seconds. This value cannot change while the authorization state machine is operating.
docsBpiCmRekeyWaitTimeout		Rekey wait timeout, in seconds. This value cannot change while the authorization state machine is operating.
docsBpiCmAuthRejectWaitTimeout		Authorization reject wait timeout, in seconds. This value cannot change while the authorization state machine is operating.
docsBpiCmAuthRequests		Number of times the cable modem has transmitted an authorization request message.
docsBpiCmAuthReplies		Number of times the cable modem receives an authorization reply message.
docsBpiCmAuthRejects		Number of times the cable modem receives an authorization reject message.
docsBpiCmAuthInvalids		Number to times the cable modem receives an authorization invalid message.
docsBpiCmAuthRejectErrorCode The options are:		Enumerated description of the error code in the most recent authorization reject message the cable modem receives.
	none	No authorization reject messages have been received since reboot.
	unknown	Last error code value was zero.
	unauthorized cm	The cable modem received an Authorization Reject in response to an Authorization Request with an error code of 1 (unauthorized cable modem).
	unauthorized SID	The cable modem received an Authorization Reject in response to an Authorization Request with an error code of 2 (unauthorized SAID).
docsBpiCmAuthRejectErrorString		The display string in the most recent authorization reject message the cable modem receives since reboot. If no authorization has been received, this value is zero.

CLI Output		Description
docsBpiCmAuthInvalidErrorCode		Enumerated description of the error code in the most recent authorization invalid message that the cable modem receives.
	none	No authorization invalid messages have been received since reboot.
	unknown	Last error code value was zero.
	unauthorized cm	The cable modem received an Authorization Invalid message from the CMTS with an error code of 1 (unauthorized cable modem). This indicates that the CMTS and cable modem have lost authorization key synchronization.
	unauthorized SID	The cable modem received a Key Reject with an error code of 2 (unauthorized SAID).
docsBpiCmAuthInvalidErrorString		Display string in most recent Authorization Invalid message received by the cable modem.

Table 21-12 docsBpiCmTEKTable Parameters

CLI Output	Description
docsBpiCmTEKPrivacyEnable	Identifies if this SID is provisioned to run Baseline Privacy.
docsBpiCmTEKState	State of the indicated TEK FSM. The options are: ■ Start ■ OPWait ■ OpReauthWait ■ Operational ■ Rekey Wait ■ Rekey Reauth Wait
docsBpiCmTEKExpiresOld	Actual clock time for expiration of the immediate predecessor of the most recent TEK for this FSM.
docsBpiCmTEKExpiresNew	Actual clock time for expiration of the most recent TEK for this FSM.
docsBpiCmTEKKeyRequests	Number of times the cable modem transmits a key request message.
docsBpiCmTEKKeyReplies	Number of times the cable modem receives a key reply message, including a message whose authentication fails.
docsBpiCmTEKKeyRejects	Number of times the cable modem receives a key reject message, including a message whose authentication fails.
docsBpiCmTEKInvalids	Number of times the cable modem receives a TEK invalid message, including a message whose authentication fails.
docsBpiCmTEKAuthPends	Number of times an authentication pending event occurs in this FSM.
docsBpiCmTEKKeyRejectErrorCode	Enumerated description of the error-code in most recent key reject message received by the cable modem.
none	No key reject message has been received since reboot.
unknown	Last error-code was zero.
unauthorized SID	SID was unauthorized.

CLI Output	Description
docsBpiCmTEKKeyRejectErrorCode	Display string in the most recent key reject message received by the cable modem. This displays a zero length string if no key reject message has been received since reboot.
docsBpiCmTEKKeyRejectErrorString	Display string in most recent key reject message received by the cable modem.
docsBpiCmTEKInvalidErrorCode	Enumerated description of the error code in the modem recent TEK invalid message reviewed by the cable modem.
None	No TEK invalid message has been received since reboot.
unknown	Last error code was zero.
invalid key sequence	Invalid key sequence.
docsBpiCmTEKInvalidErrorString	Display string in the most recent TEK invalid message received by the cable modem. If no TEK invalid message has been received since reboot, this value displays as a zero length string.

Table 21-13 docsBpi2CmBaseTable Parameters

CLI Output		Description
docsBpi2CmPrivacyEnable		Indicates if the cable modem is provisioned to run Baseline Privacy Plus.
docsBpi2CmPublicKey		Indicates the DER-encoded RSAPublicKey ASN.1 type string, as defined in the RSA Encryption Standard (PKCS #1) [10], corresponding to the public key of the cable modem. The 74, 106, 140, 204, and 270 byte key encoding lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit, and 2048 public moduli respectively.
docsBpi2CmAuthState The options are:		State of the cable modem authorization Finite State Machine (FSM).
	Start	FSM is in its initial state.
	authwait	The cable modem has received the "Provisioned" event, indicating that it has completed RF MAC registration with the CMTS. In response to receiving the event, the cable modem has sent both an Authentication Information and an Authorize Request message to the CMTS and is waiting for the reply.
	Authorized	The cable modem has received an Authorization Reply message which contains a list of valid SAIDs for this cable modem. At this point, the modem has a valid Authorization Key and SAID list. Transition into this state triggers the creation of on TEK FSM for each of the cable modem's privacy-enabled SAIDs.
	ReauthWait	The cable modem has an outstanding re-authorization request. The cable modem was either about to time out its current authorization or received an indication (an Authorization Invalid message from the CMTS) that its authorization was no longer valid. The cable modem sent an Authorization Request message to the CMTS and is waiting for a response.

CLI Output		Description
	Auth Reject Wait	The cable modem received an Authorization Reject message in response to its last Authorization Request. The Authorization Reject's error code indicated that the error was not of a permanent nature. In response to receiving this reject message, the cable modem set a timer and transitioned to the Authorized Reject Wait state. The cable modem remains in this state until the timer expires.
	Silent	The cable modem received an Authorization Reject message in response to its last Authorization Request. The Authorization Reject's error code indicated the error was of a permanent nature. This triggers a transition to the Silent state, where the cable modem is not permitted to pass CPE traffic, but is able to respond to SNMP management requests arriving from across the cable network.
docsBpi2CmAuthKeySequence Number		Most recent authorized key sequence number of this FSM.
docsBpi2CmAuthExpiresOld		Actual clock time for expiration of the immediate predecessor of the most recent authorization key for this FSM. If this FSM has only one authorization key, then the value is the time of activation of this FSM.
docsBpi2CmAuthExpiresNew		Actual clock time for expiration of the most recent authorization key for this FSM.
docsBpi2CmAuthReset		Determines the reauthorize event status.
The options are:		
	True	Generates a reauthorize event in the authorization FSM
	False	Does not generate an authorization event.
docsBpi2CmAuthGraceTime		Grace time for an authorization key. A cable modem is expected to start trying for a new authorization key beginning the grace time number of seconds before the authorization key actually expires.

CLI Output	Description
docsBpi2CmTEKGraceTime	TEK Grace Time in seconds before TEK expires.
docsBpi2CmAuthWaitTimeout	Retransmission interval, in seconds, of Authorization Request messages from the Authorize Wait state.
docsBpi2CmReauthWaitTimeout	Retransmission interval, in seconds, of Authorization Request messages from the Authorize Wait state.
docsBpi2CmOpWaitTimeout	Retransmission interval, in seconds, of Key Requests from the Operational Wait state.
docsBpi2CmRekeyWaitTimeout	Retransmission interval, in seconds, of Key Requests from the Rekey Wait state.
docsBpi2CmAuthRejectWaitTimeou	Amount of time a CM waits (seconds) in the Authorize Reject Wait state after receiving an Authorization Reject.
docsBpi2CmSAMapWaitTimeout	Retransmission interval, in seconds, of SA Map Requests from the MAP Wait state.
docsBpi2CmSAMapMaxRetries	Maximum number of Map Request retries allowed.
docsBpi2CmAuthentInfos	Number of times the CM has transmitted an Authentication Information message.
docsBpi2CmAuthRequests	Number of times the cable modem has transmitted an authorization request message.
docsBpi2CmAuthReplies	Number of times the cable modem has receive an authorization reply message.
docsBpi2CmAuthRejects	Number of times the cable modem has received an authorization reject message.
docsBpi2CmAuthInvalids	Number of times the cable modem has received an authorization invalid message.
docsBpi2CmAuthRejectErrorCode The options are:	Enumerated description of the error code in the most recent authorization reject message the cable modem receives.
none	No authorization reject messages have been received since reboot.
unknown	Last error code value was zero.

CLI Output		Description
	unauthorized cm	The cable modem received an Authorization Reject in response to an Authorization Request with an error code of 1 (unauthorized cable modem).
	unauthorized SAID	The cable modem received an Authorization Reject in response to an Authorization Request with an error code of 2 (unauthorized SAID).
	permanent Authorization Failure	Permanent authorization failure, which indicates a number of different error conditions affecting the BPKM authorization exchange including: ■ Unknown manufacturer (CMTS does not have the CA certificate) belonging to the issuer of a CM certificate. ■ CM certificate has an invalid signature. ■ ASN.1 parsing failure during verification of CM certificate. ■ CM certificate is on the "hot list". ■ Inconsistencies between certificate data and data in accompanying BPKM attributes. ■ CM and CMTS have incompatible security capabilities.
	timeOfDay NotAcquired	Time of day not acquired.
docsBpi2CmAuthRejectErrorString		Display string in the most recent authorization reject message the cable modem receives since reboot. If no authorization has been received, this value is zero.
docsBpi2CmAuthInvalidErrorCode		Enumerated description of the error code in the most recent authorization invalid message that the cable modem receives.
	none	No authorization invalid messages have been received since reboot.
	unknown	Last error code value was zero.

CLI Output		Description
	unauthorized cm	The cable modem received an Authorization Invalid message from the CMTS with an error code of 1 (unauthorized cable modem). This indicates that the CMTS and cable modem have lost authorization key synchronization.
	unsolicited	Unsolicited.
	invalidkey sequence	Invalid key sequence number.
	keyRequest Authentication Failure	Message (key request) authentication failure.
docsBpi2CmAuthInvalidErrorString		Display string in most recent Authorization Invalid message received by the cable modem.

Table 21-14 docsBpi2CmTEKTable

CLI Output	Description
docsBpi2CmTEKSAType	Type of security association. The options are: ■ none ■ primary ■ static ■ dynamic
docsBpi2CmTEKData EncryptAlg	Data encryption algorithm being used. The options are: ■ none ■ des56cbcmode ■ des40cbcmode
docsBpi2CmTEKData AuthentAlg	Data authentication algorithm being used.
docsBpi2CmTEKState	State of the indicated TEK FSM. The options are:
start	The FSM is in the initial state.
opwait	The TEK state machine has sent its initial request (Key Request) for its SAID's keying material (traffic encryption key and CBC initialization vector), and is waiting for a reply from the CMTS.
opreauthwait	This is the wait state in which the TEK state machine is placed if it does not have valid keying material while the Authorization state machine is in the middle of a reauthorization cycle.
Operational	The cable modem has valid keying material for the associated SAID.
Rekey Wait	The TEK Refresh Timer has expired and the cable modem has requested a key update for this SAID. Note that the newer of its two TEKs has not expired and can still be used for both encrypting and decrypting data traffic.
Rekey Reauth Wait	This is the wait state in which the TEK state machine is placed if the TEK state machine has valid traffic keying material, has an outstanding request for the latest keying material, and the Authorization State Machine initiates a reauthorization cycle.

CLI Output	Description
docsBpi2CmTEKKey Sequence Number	Most recent TEK key sequence number for this TEK FSM.
docsBpi2CmTEKExpiresOld	Actual clock time for expiration of the immediate predecessor of the most recent TEK for this FSM.
docsBpi2CmTEKExpiresNew	Actual clock time for expiration of the most recent TEK for this FSM.
docsBpi2CmTEKKeyRequests	Number of times the cable modem transmits a key request message.
docsBpi2CmTEKKeyReplies	Number of times the cable modem receives a key reply message, including a message whose authentication fails.
docsBpi2CmTEKKeyRejects	Number of times the cable modem receives a key reject message, including a message whose authentication fails.
docsBpi2CmTEKInvalids	Number of times the cable modem receives a TEK invalid message, including a message whose authentication fails.
docsBpi2CmTEKAuthPends	Number of times an authentication pending event occurs in this FSM.
docsBpi2CmTEKKeyReject Error Code	Enumerated description of the error-code in most recent key reject message received by the cable modem.
none	No key reject message has been received since reboot.
unknown	Last error-code was zero.
unauthorized SAID	SAID was unauthorized.
docsBpi2CmTEKKeyReject Error String	Display string in the most recent key reject message received by the cable modem. This displays a zero length string if no key reject message has been received since reboot.
docsBpi2CmTEKInvalidError Code	Enumerated description of the error code in the modem recent TEK invalid message reviewed by the cable modem.
None	No TEK invalid message has been received since reboot.
unknown	Last error code was zero.
invalid key sequence	Invalid key sequence.

CLI Output	Description
docsBpi2CmTEKInvalidErrorString	Display string in the most recent TEK invalid message received by the cable modem. If no TEK invalid message has been received since reboot, this value displays as a zero length string.

Table 21-15 systemGroup Parameters

CLI Output	Description
Descriptor	Provides a textual description of the cable modem vendor.
Contact	A contact person for the network.
Name	The name of the network device.
Location	Location of the network device.

Table 21-16 docsDevBase, docsDevSoftware, and docsDevServer Parameters

CLI Output	Description
Serial Number	Manufacturer's serial number for this device.
STP Control	Controls operation of the spanning tree protocol (as distinguished from transparent bridging). Values are:
stEnabled	Spanning tree protocol is enabled, subject to bridging constraints.
noStFilterBpdu	Spanning tree is not active, and Bridge PDUs received are discarded.
noStPassBpdu	Spanning tree is not active and Bridge PDUs are transparently forwarded.
SW Server	IP address of the TFTP server used for software upgrades. If the TFTP server is unknown, 0.0.0.0 is displayed.
SW Filename	Filename of the software image to be loaded into this device. Unless set via SNMP, this is the file name specified by the provisioning server that corresponds to the software version that is desired for this device. If the filename is unknown, the string "unknown" is returned.
Admin Status	Current provisioning administrative status. Values include:

CLI Output	Description
Upgrade FromMgt	Device will initiate a TFTP software image download. After successfully receiving an image, the device will set its state to IgnoreProvisioningUpgrade and reboot. If the download process is interrupted by a reset or power failure, the device will load the previous image and, after re-initialization, continue to attempt loading the image.
Allow Provisioning Upgrade	Device will use the software version information supplied by the provisioning server when next rebooting (this does not cause a reboot). This status appears at initial startup.
Ignore Provisioning Upgrade	Device will disregard software image upgrade information from the provisioning server.
Oper Status	Current provisioning operational status. Values include:
InProgress	A TFTP download is underway, either as a result of a version mismatch at provisioning or as a result of an upgradeFromMgt request.
Complete From Provisioning	The last software upgrade was a result of version mismatch at provisioning.
Complete FromMgt	The last software upgrade was a result of setting docsDevSwAdminStatus to upgradeFromMgt.
Failed	The last attempted download failed, ordinarily due to TFTP timeout.
Other	State other than the ones described above.
Current Version	Software version currently operating in this device.
Boot State	Current boot state. Values include:
Operational	Device has completed loading and processing of configuration parameters and the CMTS has completed the Registration exchange.
Disabled	Device was administratively disabled, possibly by being refused network access in the configuration file.
WaitingFor DhcpOffer	A DHCP Discover has been transmitted and no offer has yet been received.
WaitingFor Dhcp Response	A DHCP Request has been transmitted and no response has yet been received.
WaitingFor TimeServer	A Time Request has been transmitted and no response has yet been received.

CLI Output	Description
WaitingFor Tftp	A request to the TFTP parameter server has been made and no response has been received.
RefusedBy Cmts	The Registration Request/Response exchange with the CMTS failed.
Forwarding Denied	The registration process completed, but the network access option in the received configuration file prohibits forwarding.
Other	State other than the ones described above.
Unknown	Unknown state.
DHCP Server	IP address of the DHCP server that assigned an IP address to this device. This field displays 0.0.0.0 if DHCP was not used for IP address assignment.
Time Server	IP address of the Time server (RFC-868). This field displays 0.0.0.0 if the time server IP address is unknown.
TFTP Server	IP address of the TFTP server responsible for downloading provisioning and configuration parameters to this device. This field displays 0.0.0.0 if the TFTP server address is unknown.
Server Config File	Name of the device configuration file read from the TFTP server. This field displays an empty string if the configuration file name is unknown.

Table 21-17 docsDevEvControlTable Parameters (Configuration)

CLI Output	Description
Syslog Server	IP address of the Syslog server. If the value is 0.0.0.0, the syslog transmission is inhibited.
Threshold	Number of trap/syslog events per Throttle Interval to transmit before throttling.
Interval (seconds)	Interval over which the trap threshold applies. At initial startup, this value is one.
Admin Status The options are:	Controls the transmission of traps and syslog messages with respect to the trap pacing threshold. A single event is counted as a single event for threshold counting. That is, an event causing both a trap and a syslog message is still considered a single event.
unconstrained	Causes traps and syslog messages to transmit without regard for threshold settings. At initial startup, this is the default.
maintainBelowThreshold	Causes the suppression of trap transmission and syslog messages if the number of traps would otherwise exceed the threshold.
stopAtThreshold	Causes trap transmission to cease at the threshold, and not resume until you manually intervene.
inhibited	Causes the suppression of all trap transmission and syslog messages.
Inhibited	Indicates whether the trap and syslog transmission is inhibited because of thresholds or the current settings of the Throttle Admin parameter.

Table 21-18 docsDevEventTable Parameters

CLI Output	Description
First Time	Creation time for the entry.
Last Time	If multiple events are reported through the same entry, the time that the last event for this entry occurred.
Counts	Number of consecutive event instances reported by this entry.
Level	Priority level for this event, as defined by the vendor. These are ordered from most serious (emergency) to least serious (debug). The options are: ■ emergency ■ alert ■ critical ■ error ■ warning ■ notice ■ information ■ debug
ID	Unique identifier the type of event.
Text	Description of the event.

Table 21-19 docsDevEvControlTable Parameters (Control)

CLI Output	Description
Priority	The priority level of the particular event that occurred for the particular cable modem. Priority levels are ordered from the most serious to the least serious. The priority levels are: ■ emergency ■ alert ■ critical ■ error ■ warning ■ notice ■ information ■ debug
Action	Determines how the event notification is sent. The options are: ■ local: The event logs to the internal log. ■ trap: The event logs generates a trap. ■ syslog: A syslog message is sent.

Table 21-20 ifTable and IfXTable Parameters and Statistics

CLI Output	Description
Description	Identifies the interface.
Type	Type of interface.
Admin Status	Desired state of the interface. When the CMTS initalizes, all interfaces are down. You must either manually or configure the interfaces to be in a testing state or be up.
Oper Status	Current operational state of the interface.
The options are:	
up	Interface is operating normally and is ready to pass packets.
down	Interface is not operating.
testing	No operational packets can be passed.
unknown	State of the interface cannot be determined
dormant	Interface is ready to transmit and receive network traffic.
In Octets	Total number of octets the interface receives, including framing characters.
In Unicast Packets	Number of packets, delivered to this sub layer to a higher sublayer that were not addressed to a multicast or broadcast address at this sub-layer.
In Multicast Packets	Number of packets, delivered by this sub layer to a higher sub layer that were not addressed to a multicast address at this sub layer.
In Broadcast Packets	Number of packets, delivered to this sub-layer to a higher sub layer that were not addressed to a broadcast address at this sub layer.
In Errors	Number of inbound packets that contain errors preventing them from being deliverable to a higher layer protocol.
In Discards	Number of inbound packets that were chosen to be discarded even though no errors were detected to prevent their being deliverable to a higher layer protocol. The reason for discarding the packets could be to free up buffer space.
Out Octets	Number of packets received through the interface that were discarded because of an unknown or unsupported protocol.

CLI Output	Description
Out Unicast Packets	Total number of packets that higher level protocols requested be transmitted and were not addressed to a multicast or broadcast address at this sub layer, including those that were discarded or not sent.
Out Multicast Packets	Total number of packets that higher level protocols request be transmitted and were addressed to a multicast address at this sub layer.
Out Broadcast Packets	Total number of packets that higher level protocols requested to be transmitted and were addressed to a broadcast address at this sub layer, including those that ere discarded or not sent.
Out Errors	Number of outbound packets that could not be transmitted because of errors.
Out Discards	Number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted.

Table 21-21 pktcMtaDevBaseTable Parameters

CLI Output	Description
Serial Number	Manufacturer's serial number for this MTA.
Hardware Version	Manufacturer's hardware version for this MTA.
MAC Address	Telephony MAC address for this addrss
Fully Qualified Domain Name	Fully qualified domain name for this MTA.
End Points	Physical end points for this MTA.
Voice Enabled	MTA admininistrive status for this device. The options are: ■ True: Voice is enabled ■ False: Voice is disabled
Type ID	Device type identifier for the DHCP option 60 exchanged between the MA and the DHCP server.
Provisioned State	Indicates the completion state of the provisioning process. The options are: ■ Pass: Pass state occurs after completing the processing of the configuration file. ■ In Progress: Occurs from boot time until configuration file processing is complete. ■ Fail: Pass state occurs after completing the processing of the configuration file. Manual intervention is required.
HTTP Access	Indicates whether HTTP file access is supported for MTA configuration file transfer.

Table 21-22 pktcMtaDevServerTable Parameters

CLI Output	Description
Boot State	The state of the server. The options are: ■ Operational: Device is done loading and processing configuration parameters, and the CMTS has completed the registration exchange. ■ Disabled: Device was administratively disabled, possibly by being refused network access in the configuration file. ■ waiting for Dhcp Offer: DHCP discover has been transmitted and no offer has been received. ■ Waiting for Dhcp Response: DHCP request has been trasmitted and no response has yet been received. ■ Waiting For Config: Request for configuration server has been made and no response received. ■ Refused by CMTS: Registration request/response exchanged with the CMTS failed. ■ Other: Other reason besides those listed above. ■ Unknown: The state is unknown.
DHCP Server	IP address or fully qualified domain name (FQDN) of the DHCP server that assigned an address to this device. This value is 0.0.0.0 if the DHCP derver is not used for the IP assignment address.
Primary DNS Server	IP address for FQDN of the primary DNS server that resolved an IP address for this device.
Secondary DNS Server	IP address or FQDN of the secondary DNS server that resolved an IP address for this device.
Configuration File	URL of the TFTP/HTTP file for downloading provisioning and configuration parameters to this device. This is a value of null is the server address is unknown.
SNMP Entity	IP address or FQDN of the SNMP entity for provisioning trap handling that assigned an IP address to this device. This value is 0.0.0.0 if DHCP was not used for IP address assignment.

Table 21-23 pktcSigEndPntConfigTable Parameters

CLI Output	Description
Call Agent ID	The call agent name. The call agent name can be a FQDN or an IP address.
Call Agent UDP Port	The call agent UDP port for this instance of call signalling.

Table 21-24 pktcSigDevCodecTable Parameters

CLI Output	Description
Index	Index for this codec.
Type	CODEC conversion types that are supported by the MTA: ■ g729 ■ g729a ■ g729e ■ g711mu ■ g726 ■ g728

Table 21-25 MTA Service-level Configuration Parameters

CLI Output	Description
Echo Cancellation	Displays whether echoes are cancelled (True or False). True indicates that echo cancellation is in use. False indicates that echo cancellation is not in use.
Silence Suppression	Displays whether silence is suppressed in the send direction (True or False). True indicates that silence suppression is enabled. False indicates that silence suppression is disabled.
Connection Mode	Displays the various ways in which the MTA can connect to the network (such as voice, fax, and modem).
R0 Cadence	Displays ring cadence intervals, where each bit represents a duration of 200 milliseconds (6 second total).
R6 Cadence	Displays ring cadence intervals, where each bit represents a duration of 200 milliseconds (6 second total).
R7 Cadence	Displays ring cadence intervals, where each bit represents a duration of 200 milliseconds (6 second total).
Def Call Signal TOS	Displays the default Type of Service (TOS) value for call signalling (signals for setting up calls) in the IP header.
Def Media Stream TOS	Displays the default Type of Service (TOS) value for media stream packets in the IP header. Audio and video streams are examples of media streams.

CLI Output	Description
TOS Format Selector	Displays one of the following formats for the default call signalling and media stream TOS values: ■ dscpCodepoint – Specifies that the TOS field is treated as a Differentiated Service Code Point (DSCP). The TOS field in the IP header identifies the differentiated service per hop behavior, which enables intermediate routers to select packets and apply specific forwarding rules based on the value of the TOS byte. ■ ipv4TOSOctet – Specifies that the TOS field is treated as an IPv4 TOS octet. Networks can provide a specific level of service based on the octet value in the packet.

A COMMAND SUMMARY

This chapter provides a summary of all Cuda 12000 CLI-based administration commands categorized by primary function. Note that the **no** and **show** forms of the commands are not included.

The following command functions are listed in this chapter:

Access Control Commands

Table A-1 Access Control Commands

Command	Mode	Description
aaa authentication login default {local | tacacs+ | radius}	root	Enables RADIUS and TACACS+ access authentication.
access-profile <profile name> description <text string> {addprivilege | removeprivilege} {admin | hfc | observer | prov | router} {noaccess | read/write | readonly}	root	Creates access profiles, which are are applied to user accounts to define the functional areas accessible to the user and the access rights (read/write) for those areas.
account <account name> [add-profile | remove-profile <profile> [password <password>] [description <string>]	root	Creates or modifies user accounts; apply access profiles and privileges to users.
enable <username>	root	Used to log into the CLI environment.
quit	Any	Exits out of the Cuda 12000 CLI environment.
radius-server {host <ip-address | key {<number> | <string>}}	root	Specifies a RADIUS authentication server.
set password <new-password>	Any	Changes the password of the current account.
tacacs-server {host <ip-address | key <string>}	root	Sets the IP address and encryption key of the TACACS+ authentication server.

Mode Commands

Table A-2 Mode Commands

Command	Mode	Description
interface <c/s/i>	Any	Enters configuration mode for a selected interface.
ip address <ip-address> <mask>	interface <c/s/i>	Enters IP interface mode.
prov-server	Any	Enters provisioning server mode. This command applies only if the FastFlow Broadband Provisioning Manager is installed on the Cuda 12000.
root	Any	Enters root mode from within any mode.
router	Any	Enters router mode from which you can access RIP and OSPF configuration modes.
router ospf	Any	Enters OSPF global configuration mode.
router rip	Any	Enters RIP configuration mode.
slot <c/s>	Any	Enters configuration mode for a selected slot.
up	Any	Moves you back up one mode level.

General Commands

Table A-3 General Commands

Command	Mode	Description
help <command>	Any	Shows all commands available in the current mode, and provides a brief description of each. Note that you can enter <command> **?** to see a list of available commands without their associated descriptions.
set paging {on | off}	Any	Enables or disables the paging of display output.
set prompt [mode]	Any	Sets the prompt mode. You can choose to show the IP address and current mode within the prompt display; or choose to display no information within the prompt.
set time <string>	Any	Sets the system time.
set timeout <minutes>	Any	Sets the timeout for idle CLI sessions.
shutdown	interface:<type>:csi	Disables a specified interface.
talk	Any	Enables sending of broadcast messages to users. This command also allows you to send a broadcast message.

IP Administration and Route Filtering Commands

Table A-4 IP Administration and Route Filtering Commands

Command	Mode	Description
access-class <list number> {in | out} priority <priority number>	interface:cable:csi	Applies filtering rules (access-lists) to interfaces.
access-list <list number> {deny | permit} <rule number> ip {<source ip address> <source IP mask> | host <ip address> | any } {<destination IP address> <destination IP mask> | host <destination ip address> | any} [tos <tos> <tos mask>] [change-tos <tos>]	■ root ■ interface:cable:csi	A sequential set of filtering rules to which inbound and outbound packets can be applied.
add arp <ip-address> <mac-address>	interface:<type>:csi	Adds an ARP (Address Resolution Protocol) entry to the ARP cache on a selected interface.
arp timeout <number>	interface:<type>:csi	Sets the timeout for dynamic ARP cache entries.
clear arp-cache	■ root ■ interface:<type>:csi	Deletes all non-static ARP entries.
clear ip igmp group [<group-address>]	■ root ■ interface:<type>:csi	Deletes multicast group entries.
del arp <ip-address>	interface:<type>:csi	Removes an ARP (Address Resolution Protocol) entry from the ARP cache on a selected interface.
dhcp-authority {enable | disable}	interface:<type>:csi	Enables and configures DHCP authority ranges on the current interface. DHCP authority secures all IP addresses Any address that falls within the range are labelled as DHCP within the ARP table upon acknowledgement from the host. This tagging of the ARP cache entry ensures that another MAC host cannot use the IP address.
export	■ router:rip ■ router:ospf	Enters export mode. From within this mode you can create export Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) route filters.

Command	Mode	Description
filter-aging {in | out} {enable | disable | rate <seconds}	slot	Sets IP packet filter aging for the module within the current slot.
import	■ router-rip ■ router-ospf	Enters import mode. From within this mode you can create import Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) route filters.
ip address <ip-address> <mask> [{other | primary | secondary}]	interface:<type>:csi	Adds an IP interface *(*IP address*)* to the current physical interface. The system supports both primary and secondary addresses.
ip filter {in | out} {enable | disable}	interface:<type>:csi	Enables or disables IP packet filtering *(using access lists)* on the current interface.
ip igmp {join-group <group-address> | query-interval <seconds> | query-max-response-time <seconds> | version {2 | 1 | v2_only} | robustness <value> | router | last-query-interval <seconds>}	interface:<type>:csi	Configures the interface for IGMP communications.
ip igmp proxy <group-address> <group-mask> metric <number> <c/s/i>	root	Configures interfaces to act as IGMP proxies for a single multicast group or a range of multicast groups.
ip route <dest-network> <mask> <gateway-ip-address> [<metric>]	interface:<type>:csi	Adds a static route to the current interface. You can use this command to add both local and remote routes.
ip route default <gateway-ip-address> [<metric>]	Any	Configures the default IP route.
ip source-route <ip address> <mask> <next hop gateway>	interface:<type>:csi	Adds a source routing entry on the current interface to ensure local destinations are used first should no default root exist for the network.
map-list {route-map-list-number} route-map	■ router:ospf:import ■ router:ospf:export ■ router:rip:import ■ router:rip:export	Adds a route-map to a specified map-list for creation of RIP and OSPF route filters.

Command	Mode	Description
match {ip-address <ip-address> <mask> | neighbor <ip-address> <mask> |tag <tag-value> [exact | exclude]}	■ router:ospf:import ■ router:ospf:export ■ router:rip:import ■ router:rip:export	Creates match attributes for import and export filters.
override {metric <metric-value> | tag <tag-value>	■ router:ospf:import ■ router:ospf:export ■ router:rip:import ■ router:rip:export	Defines override rules for import or export filters.
ping [-l <size>] [-n <count>] [-w <timeout>] [-s <src-ip-address>] <dest-ip-address>	Any	Sends ICMP echo request packets to test network device availability.
proxy-arp	interface:cable:csi	Enables proxy ARP on the current cable interface.
route-map <map-tag> {permit | deny}	■ router:ospf:import ■ router:ospf:export ■ router:rip:import ■ router:rip:export	Defines route-maps for use in map-lists for route filtering.
traceroute [-w <timeout>] [-i <number>] [-m <number>] [-p <number>][-q <number>] [-t <number>] [-s <src-ip-address>] [-x <number>] [-F] <dest-ip-address> [<data-size>]	Any	Traces the route that packets traverse from the Cuda 12000 to a destination.

RIP Commands

Table A-5 Routing Information Protocol (RIP) Commands

Command	Mode	Description
ip rip accept default-route	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the interface to accept default routes advertised by neighbor routers.
ip rip accept host-route	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the interface to accept host routes advertised by neighbor routers.
ip rip authentication {md5 | password}	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the authentication type for the RIP interface.
ip rip authentication {key-id <id> key <key> | key <key>}	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the authentication key for the RIP interface.
ip rip cost <number>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Enters the cost or metric of the current RIP interface.
ip rip default cost <number>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Enters the cost or metric that is to be used for the default route entry in RIP updates originated on the RIP interface.
ip rip disable	interface:<type>:csi <c/s/i>:ip-address <ip address>	Disables RIP on the current IP interface.
ip rip enable	interface:<type>:csi <c/s/i>:ip-address <ip address>	Enables RIP on the current IP interface.
ip rip neighbor <ip-address>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the IP addresses of RIP neighbors.
ip rip poisoned-reverse	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the interface to implement poison reverse.
ip rip receive-version {1|2|1 2| none}	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the version of RIP the current IP interface uses to receive routes.
ip rip send default-also	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the interface to advertise the default route in addition to other routes.

Command	Mode	Description
ip rip send default-only	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the interface to advertise only the Cuda 12000's default route.
ip rip send-version {1 | 2 |1 2| none}	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the version of RIP the current IP interface uses to advertise routes.
ip rip split-horizon	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the interface to implement split horizon.
reset rip stats	root	Resets RIP statistical counters.
rip	router	Enters RIP configuration mode for access to import and export filter configuration.
router rip	Any	Enters RIP configuration mode for access to import and export filter configuration.

OSPF Commands

Table A-6 Open Shortest Path First (OSPF) Commands

Command	Mode	Description
asbr	router:ospf	Configures the system as an Autonomous System Border Router (ASBR).
ip ospf area-id <area-id> [{enable | disable}]	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the Area ID for the current interface. The Area ID designates the OSPF area to which this IP interface belongs.
ip ospf authentication {md5 | password}	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the authentication type for the OSPF interface.
ip ospf authentication {key-id <id> key <key> | key <key>}	interface:<type>:csi <c/s/i>:ip-address <ip address>	Configures the authentication key for the OSPF interface.
ip ospf cost <number>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Defines the cost metric for this IP interface.
ip ospf dead-interval <number of seconds>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Defines the dead-interval for this IP interface.
ip ospf hello-interval <number of seconds>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Specifies the length of time, in seconds, the router waits between sending Hello packets on the current IP interface.
ip ospf priority <number>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Assigns the interface priority the current IP interface.
ip ospf retransmit-interval <number of seconds>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Sets the Link State Advertisement (LSA) retransmit interval in seconds for the current IP interface.
ip ospf transit-delay <number of seconds>	interface:<type>:csi <c/s/i>:ip-address <ip address>	Sets the number of seconds it takes to transmit a link state update packet on the current IP interface.

Command	Mode	Description
ospf area <area-id> <area id> [authentication {md5 | password}] [[stub [no-summary]] [default-cost <cost>] [range <ip-address> <mask> [advertise-matching]] [{enable | disable}]	router:ospf	Creates an OSPF area which you can then apply to select interfaces using the ip ospf area-id command within interface:<type>:csi mode.
ospf-vi <transit-area-id> <neighbor-router-id>	router:ospf	Configures the transit area and the router ID of the neighbor for a virtual interface.
ospf-vi <transit-area-id> <neighbor-router-id> authentication {md5 | password}	router:ospf	Configures the authentication type for a virtual interface.
ospf-vi <transit-area-id> <neighbor-router-id> authentication {key-id <id> key <key> | key <key>}	router:ospf	Configures the authentication key for a virtual interface.
ospf-vi <transit-area-id> <neighbor-router-id> dead-interval <number of seconds>	router:ospf	Configures the dead-interval for a virtual interface.
ospf-vi <transit-area-id> <neighbor-router-id> hello-interval <number of seconds>	router:ospf	Configures the Hello interval for a virtual interface.
ospf-vi <transit-area-id> <neighbor-router-id> retransmit-interval <number of seconds>	router:ospf	Configures the retransmit interval for a virtual interface.
ospf-vi <transit-area-id> <neighbor-router-id> transit-delay <number of seconds>	router:ospf	Configures the transit delay for a virtual interface.
report {ospf-nbr-state | ospf-virt-nbr-state}	router:ospf	Enables sending of OSPF neighbor state trap and OSPF virtual neighbor state trap.
router ospf	Any	Enters OSPF global configuration mode.
router-id <ip address>	router:ospf	Configures the OSPF router ID for this system. The router ID applies to all OSPF interfaces; a single source router ID.

DHCP Relay Commands

Table A-7 DHCP Relay Commands

Command	Mode	Description
bootp-policy <index> {deny mac <mac-address> [mask <mask>]... | permit <ip-address>... mac <mac-address> [mask <mask>]... } [description <string>]	interface:cable:csi	Configures BOOTP policies on the current interface.
dhcp-policy {<policy-number> | default} {deny | permit} {<ip-address> ... | forward-internal [disable]} [agent-option {cm | cpe} | cmmac <mac-address> | interface <c/s/i> | mac <mac-address> [mask <mask>]] [vendor-class-id {cm | mta}] [description <string>]	interface:cable:csi	Configures DHCP policies on the current interface. DHCP policies dictate the servers to which DHCP requests are forwarded.
dhcp-relay {enable | disable} [cm-gateway <gi-address>] [cpe-gateway <gi-address>] [add-agent-options {enable | disable}] drop-mismatch {enable | disable}] [relay-mode {append | replace | untouch | discard}] [max-pkt-len <number>] [mta-gateway <gi-address>] [server <gi-address>]	interface:cable:csi	Configure DHCP relay options on the current cable interface. Note that you configure the DHCP relay server using the dhcp-policy command.

Cable Interface Administration Commands

Table A-8 Cable Interface Administration Commands

Command	Mode	Description
admission-control {enable | disable}	interface:cable:csi	Enables and disables the admission control function.
cm modify upstream <new-upstream-channel> {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Moves the specified cable modem to a new upstream port. Note that the new upstream port must be enabled before issuing this command.
downstream frequency <freq-number>	interface:cable:csi	Sets the downstream frequency on the current cable (CMTS) interface.
downstream interleave-depth <number>	interface:cable:csi	Sets the downstream interleave depth on the current cable (CMTS) interface.
downstream modulation {qam64 | qam256}	interface:cable:csi	Defines the modulation type that you want the downstream channel on the current cable (CMTS) interface to use — 64qam or 256qam.
downstream no shutdown	interface:cable:csi	Sets the downstream channel status on the current cable (CMTS) interface to *up*.
downstream shutdown	interface:cable:csi	Sets the downstream channel status on the current cable (CMTS) interface to *down*.
downstream transmit-power <number>	interface:cable:csi	Sets the transmit power of the downstream channel on the current cable (CMTS) interface.
flap-list aging <value>	interface:cable:csi	Defines the number of days to retain flapping activity on cable modems connected to this cable interface.
flap-list clear	interface:cable:csi	Resets (clears) all counters in the cable modem flap list on the current cable (CMTS) interface.

Command	Mode	Description
flap-list insertion-time <value>	interface:cable:csi	Sets the cable-flap list insertion time. When a cable modem makes an insertion request more frequently than the amount of insertion time defined by this command, the system adds the cable modem to the flap list for activity recording.
flap-list power-adj-threshold <value>	interface:cable:csi	Sets the flap list power-adjustment threshold.
flap-list size <value>	interface:cable:csi	Sets the maximum number of entries that the system can record in the flap list. When the number of entries in the flap list exceeds the number that you set with this command, the oldest flap list entries are aged out.
insertion-interval <centiseconds>	interface:cable:csi	Sets the maximum amount of time that a cable modem can request an upstream frequency on this CMTS interface. This initial request is known as *ranging*.
map-timer <value>	interface:cable:csi	Sets the map timer interval, in microseconds, on a specific cable interface.
modulation-profile <profile number> interval-usage {initial | long | request | short | station} [fec-tbytes <number>] [fec-len <number>] [burst-len <number>] [mod {16qam | qpsk}] [scrambler] [no scrambler] [diff] [no diff] [seed <number>] [pre-len <number>] [last-cw {fixed | shortened}]	interface:cable:csi	Creates a cable modulation profile.
periodic-ranging-interval <value>	interface:cable:csi	Specifies how often this cable interface periodically invites modems to range.
plant-delay <value>	interface:cable:csi	Specifies the estimated plant propagation delay, in microseconds.
pll-state	interface:cable:csi	Sets the phase lock loop state (PLL) for the current cable interface.

Command	Mode	Description
qos permission modems	interface:cable:csi	Enables cable modem registration access to the QoS tables on the current cable interface.
ranging-attempts <value>	interface:cable:csi	Maximum number of ignored ranging invitations allowed by this cable interface.
shared-secret [ascii] <string>	interface:cable:csi	Use this command to configure cable modem authentication by defining the CMTS shared secret. The shared secret is defined as a hex string. Any modem that wants to register with the CMTS must return this known string *(the shared secret)*.
sync-interval <value>	interface:cable:csi	Configures the synchronization interval for the current cable interface. Valid range: 1 – 200 milliseconds.
ucd-interval <value>	interface:cable:csi	Configures the UCD interval for the current cable interface. Valid range: 1 – 2000 milliseconds.
upstream <port number> channel-width {200 | 400| 800 | 1600 | 3200}	interface:cable:csi	Sets the channel width, in kHz, for the specified upstream port.
upstream <port number> data-backoff <start value> <end value>	interface:cable:csi	Sets a fixed start value for initial backoff on the upstream channels.
upstream <port number> frequency <value>	interface:cable:csi	Sets the upstream signal frequency.
upstream <port number> map {init-maint-size-adjust <value> | max-ranging-invitations <value> | min-req-region <value> | ucd-grant-size <value>}	interface:cable:csi	Tunes map generation for the specified upstream port.
upstream <port number> minislot-size <value>	interface:cable:csi	Sets the number of 6.25 microsecond ticks in each upstream minislot.
upstream <port number> modulation-profile <profile index>	interface:cable:csi	Specifies the modulation profile that you want the port to utilize.
upstream <port number> no shutdown	interface:cable:csi	Sets the channel status for the specified upstream port to *up*.

Command	Mode	Description
upstream <port number> power-level <power level>	interface:cable:csi	Sets the receive power level for the upstream interface in Tenth dBmV.
upstream <port number> range-backoff <start range> <end range>	interface:cable:csi	Sets the range backoff on the specified upstream ports.
upstream <port number> ranging {init-range-timeout <number> | max-power-adjust <number> | power-offset-threshold <number> | zero-frequency-adjust {disable|enable} | zero-power-adjust {disable|enable}| zero-timing-adjust {disable|enable}}	interface:cable:csi	Tune how cable modems adjust power levels during the ranging process on the specified upstream port.
upstream <port number> shutdown	interface:cable:csi	Sets the channel status for the specified upstream port to *down*.
upstream <port number> voice-bw-reserve <number>	interface:cable:csi	Reserves a percentage of upstream bandwidth for voice traffic on the current interface.

Cable Modem and Subscriber Administration Commands

Table A-9 Cable Modem and Subscriber Administration Commands

Command	Mode	Description
clear service-flow log <all>	interface:cable:csi	Clears Service Flow logs.
cm-filter <group-number> <filter-number> {deny | permit} prot {any | tcp | udp | <number>} [src <ip-address> <mask>] [dest <ip-address> <mask>] [tos <tos-value> <mask>] [src-port {any | <number>}] [dest-port {any | <number>}] [tcp-flag {ack | fin | push | reset | syn | urgent} tcp-flag-mask {ack | fin | push | reset | syn | urgent}]	Any	Creates a packet filter for upstream or downstream cable modem or CPE traffic.
cm-filter-default cm downstream <group-id>	Any	Specifies the default downstream filter group for cable modems.
cm-filter-default cm upstream <group-id>	Any	Specifies the default upstream filter group for cable modems.
cm-filter-default cpe downstream <group-id>	Any	Specifies the default downstream filter group for CPE devices.
cm-filter-default cpe upstream <group-id>	Any	Specifies the default upstream filter group for CPE devices.
cm-offline clear	interface:cable:csi	Flushes all offline modems on the current interface from the CMTS tables.
cm-offline persist	interface:cable:csi	Enables the CMTS to maintain statistics for a cable modem after the modem goes offline.
cm-offline timer <number>	interface:cable:csi	Specifies the number of days that the CMTS tracks offline cable modems.
cm cpe-reset {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Clears the CPE IP addresses that the CMTS has learned for the cable modem.
cm modify active {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Enables the ability of the CMTS to perform subscriber management of devices associated with the specified cable modem.

Command	Mode	Description
cm modify cm-downstream <group-id> {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Assigns a downstream filter group to a cable modem.
cm modify cm-upstream <group-id> {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Assigns an upstream filter group to a cable modem.
cm modify cpe-downstream <group-id> {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Assigns a downstream filter group to CPE devices that access the network through the specified cable modem.
cm modify cpe-upstream <group-id> {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Assigns an upstream filter group to CPE devices that access the network through the specified cable modem.
cm modify learnable {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Enables the ability of the CMTS to discover IP addresses of CPE associated with the specified cable modem.
cm modify max-ip <number> {<ip-address> | <mac-address> | <sid>}	interface:cable:csi	Sets the maximum number of IP addresses of CPE devices associated with the specified cable modem that can access the network.
cm reset	interface:cable:csi	Resets cable modems attached to the current cable interface.
cpe-control active	Any	Specifies whether the CMTS performs subscriber management by default for cable modems.
cpe-control learnable	Any	Specifies whether the CMTS discovers the IP address of CPE automatically.
cpe-control max-ip <number>	Any	Specifies the default maximum number of CPE IP addresses that can access the network through a single cable modem.
privacy auth <mac-address> {cm-lifetime <number> | cm-reset {invalidateAuth | invalidateTeks | sendAuthInvalid}}	interface:cable:csi	Sets the lifetime in seconds that the CMTS assigns to an authorization key for a specified cable modem.
privacy base auth-lifetime <0..6048000>	interface:cable:csi	Sets the lifetime in seconds that the CMTS assigns to a new authorization key.

Command	Mode	Description
privacy base cert-trust {trusted | untrusted}	interface:cable:csi	Sets the trust for all new self-assigned manufacturer certificates.
privacy base enable-cert-validity-periods {true | false}	interface:cable:csi	Sets the certificates to have or not to have the validity period checked against the current time of day.
privacy base tek-lifetime <0..6048000>	interface:cable:csi	Sets the lifetime in seconds that the CMTS assigns to a new TEK.
privacy ca-cert <number> [{trusted | untrusted | chained | root}] certificate <filename>	■ root ■ interface:cable:csi	Specifies manufacturer certification authority (CA) X.509 certificates.
privacy cm-cert <mac> [{trusted | untrusted}] certificate <filename>	■ root ■ interface:cable:csi	Assigns an X.509 CM certificate to a cable modem.
privacy encryption {40-bit-des | 56-bit-des}	interface:cable:csi	Specifies the type of encryption used for baseline privacy.
privacy multicast ip <index> <multicast-ip-address> <mask> said <number> sa-type {dynamic | none | primary | static} encrypt-alg {des40cbcMode | des56cbcMode | none} authent-alg none	interface:cable:csi	Sets the IP multicast address mapping entry and its prefix for the associated SAID.
privacy multicast mac <mac-address> said <number>	interface:cable:csi	Sets the MAC address of the cable modem that is authorized to access the defined multicast stream via the SAID.
privacy tek <said> {tek-lifetime <number> | reset}	interface:cable:csi	Sets the lifetime, in seconds, that the CMTS assigns to a Traffic Encryption Key (TEK) for an associated SAID.

Network-Layer Bridge Commands

Table A-10 Network-Layer Bridge Commands

Command	Mode	Description
bridge-group <string>	■ root ■ interface:cable:csi	Creates a network-layer bridge group. Network layer bridging is especially useful in spanning a single subnet across multiple modules.
bridge-interface <c/s/i>	interface:bridge-group	Adds an interface to the current bridge group.
bridge-timeout {aging <number | reply <number>}	■ interface:bridge-group ■ interface:cable:csi	Configures aging and reply timers for bridge group broadcast flows.

Fault Management Commands

Table A-11 Fault Management Commands

Command	Mode	Description
alarm-throttle {alarms <number> | interval <number> | default}	root	Configures alarm threshold and delivery parameters.+
aux-device ac-monitor <args> aux-device db15 alarm <args> aux-device dc-monitor <args> aux-device fan-rotation <args> aux-device fan-temp <args> aux-device ps-temp <args>	root	Configures an external device (e.g., power supply and fan tray) for fault reporting.
basmonitor	Any	Monitors agent-level activity.
chassis-fault [backplane] [backplane-power] [backplane-power-a] [backplane-power-b] [backplane-temp] [bits-a] [bits-b] [blue] [fan-rotation] [fan-temp] [local-pwr-a] [local-pwr-b] [processor-temp] [ps-ac] [ps-dc] [ps-temp] [red-alarm] [yellow]	root	Enables chassis alarms.
event-config reporting {default | {{emergency | alert | critical | error | warning | notice | info | debug} {none | local | local|traps | local|syslog | local|syslog|traps}}	root	Assigns an event class to a reporting action.
event-config syslog <ip-address>	root	Specifies the IP address of the syslog server.
event-config throttle admin {unconstrained | maintainBelowThreshold | stopAtThreshold | inhibited}	root	Specifies the event administrative status.
event-config throttle interval <number>	root	Specifies the event interval.
event-config throttle threshold <number>	root	Specifies the event threshold.
event-log clear	root	Clears the event log.
link-trap	interface:<type>:csi	Enables link up and link down traps for an interface.
trace-log	slot or interface:cable:csi	Enables you to trace agent-level activity through basmonitor.

Chassis Commands

Table A-12 Chassis Commands

Command	Mode	Description
aux-device backplane-clock-a {bits-a | bits-b | internal | none | slot <c/s> {enable | disable}}	root	Configures primary clock (A).
aux-device-backplane-clock-b {bits-a | bits-b | internal | none | slot <c/s> {enable | disable}}	root	Configures secondary clock (B).
boot {enabled | disabled | reset} slot <c/s>	slot	Use this command to hard boot the current module or disable the module *(bring it offline).*
ccdown	root	Shuts down the management module. *Use this command only if you have physical access to the system.*
chassis {description <string> | group <group-name> | mcs {enable | disable}}	root	Configures chassis group support.
chassis-config <chassis-number>	root	Configures the chassis ID for the selected chassis. Valid range: 1–128.
chassis-config <chassis-number> clusterid <number>	root	Assigns the selected chassis to a cluster.
chassis-config <chassis-number> manager secondary	root	Configures the management module as a primary or secondary manager.
chassis-config <chassis-number> scope {chassis | cluster}	root	Defines the management scope.
connect <ip-address> [password <password>] [user <username>]	root	Connects you to another Cuda 12000.
db-check	root	Validates provisioning database access information.
db-connect <ldap-server-object-name> <ldap-server-ip-address> <ldap-server-port-number> <ldap-server-username> <ldap-server-password>	root	Configures provisioning information.

Command	Mode	Description
http-server {enable | disable}	root	Enables and disables the HTTP server on the chassis.
lookup {enable | disable}	root	Enables the Jini lookup service on the chassis, which is required for chassis group support.
reset [{hard | soft}]	slot c/s	Resets (reboots) the module that is installed in the selected slot.
save	slot	Persists the configuration of a module in the current slot, or in all chassis slots.
traffic-relay {dns | ftp | http | snmp | snmp-trap | ssh | syslog | telnet | tftp | time_of_day} [port <port>]	root	Configures processes, such as the HTTP server, to send and receive TCP or UDP packets using an internal address on the chassis.

SNMP Commands

Table A-13 SNMP Commands

Command	Mode	Description
snmp-server community <community-name> <security-name> [address <ip-address> [mask <ip mask>]] [context <context>] [storage {volatile | nonvolatile | permanent | readonly}]	root	Creates an SNMP community.
snmp-server contact <contact>	root	Specifies the SNMP contact for the network.
snmp-server context <context-name> [storage {volatile | nonvolatile | permanent | readonly}] [status {enable | disable}]	root	Creates an SNMPv3 context.
snmp-server group <group-name> {v1 | v2c | v3 {auth | noauth | priv}} [read <readview-name>] [write <writeview-name>] [notify <notifyview-name>] [context <context-name>] [storage {volatile | nonvolatile | permanent | readonly}]	root	Creates an SNMP group.
snmp-server host <ip-address> <community-name> {traps | informs [timeout <seconds>] [retries <retries>]} [version {1 | 2c | 3 {auth | noauth | priv}}] [udp-port <port>] [mms <size>] [storage {volatile | nonvolatile | permanent | readonly}] [notification-type <type>...]	root	Creates an SNMP host to receive trap messages.
snmp-server location <location>	root	Specifies the physical location of the Cuda 12000.
snmp-server name <name>	root	Specifies the system name.
snmp-server user <user> [auth {md5 | sha} <auth-password> [priv des56 <priv-password>]] [storage {volatile | nonvolatile | permanent | readonly}] [status {enable | disable}]	root	Creates an SNMPv3 user for the SNMP entity.

Command	Mode	Description
snmp-server view <view-name> <oid-tree> {included | excluded} [storage {volatile | nonvolatile | permanent | readonly}] [status {enable | disable}]	root	Creates an SNMP MIB view.

Packet Over SONET (POS) Commands

Table A-14 POS Commands

Command	Mode	Description
clear counters	interface:pos:csi	Clears all counters on the current POS interface.
clock-source {line | internal}	interface:pos:csi	Configures the SONET transmission clock source.
crc {16 | 32}	interface:pos:csi	Configures cyclic redundancy error checking on the current POS interface.
loop {line | internal}	interface:pos:csi	Configures loopback testing on the current POS interface.
mtu <value>	interface:pos:csi	Sets the maximum transmission unit (MTU) for the interface.
pos flag {c2 | j0} <value>	interface:pos:csi	Sets values for the *c2* and *j0* SONET overhead bytes on the current POS interface.
pos report lais	interface:pos:csi	Configures the POS interface to report line alarm indication signal errors.
pos report lrdi	interface:pos:csi	Configures the POS interface to report line remote defect indication errors.
pos report pais	interface:pos:csi	Configures the POS interface to report path alarm indication signal errors.
pos report plop	interface:pos:csi	Configures the POS interface to report path loss of pointer errors.
pos report prdi	interface:pos:csi	Configures the POS interface to report path remote defect indication errors.

Command	Mode	Description
pos report sd-ber	interface:pos:csi	Configures the POS interface to report when the B2 signal degrades to meet or cross a specified Bit Error Rate (BER).
pos report sf-ber	interface:pos:csi	Configures the POS interface to report a failure when the B2 signal degrades to meet or cross a specified Bit Error Rate (BER).
pos report slof	interface:pos:csi	Configures the POS interface to report section loss of frame errors.
pos report slos	interface:pos:csi	Configures the POS interface to report loss of signal (SLOS) errors.
pos scramble	interface:pos:csi	Configures payload scrambling on the current SONET interface.
pos threshold sd-ber <number>	interface:pos:csi	Sets the threshold values associated with both the signal degrade bit error rates (BERs) alarms.
pos threshold sf-ber <number>	interface:pos:csi	Sets the threshold values associated with both the signal failure bit error rates (BERs) alarms
ppp authentication chap	interface:pos:csi	Enables authentication on the current POS interface to user CHAP authentication.
ppp authentication chap pap	interface:pos:csi	Enables the interface to negotiate the authentication protocol to use. It will try first to agree on CHAP authentication.
ppp authentication pap	interface:pos:csi	Enables the interface to use PAP authentication.
ppp authentication pap chap	interface:pos:csi	Enables the interface to negotiate the authentication protocol to use. It will try first to agree on PAP authentication.
ppp chap-hostname <name>	interface:pos:csi	Configures the username with which the POS interface responds to Challenge Handshake Protocol (CHAP) challenges.

Command	Mode	Description
ppp chap-password <password>	interface:pos:csi	Configures the password with which the POS interface responds to Challenge Handshake Protocol (CHAP) challenges.
ppp ipcp-report-address	interface:pos:csi	Configures the POS interface to respond with its IP address during Internet Protocol Control Protocol (IPCP) negotiations. Note that this is the default behavior.
ppp negotiation-count <0...100>	interface:pos:csi	Specifies the maximum number of negotiation attempts that the current POS interface allows.
ppp pap-sent-username <name> password <password>	interface:pos:csi	Configures the username and password that the POS interface sends in response to PAP authorization challenges.
ppp username <name> password <password>	interface:pos:csi	Allows you to define username/password authentication entries used by the POS interface to authenticate remote peers.
timeout <0 ... 65535>	interface:pos:csi	Sets the timeout values for both point-to-point protocol (PPP) authentication and negotiation responses.

Ethernet Commands

Table A-15 Ethernet Commands

Command	Mode	Description
duplex {auto | full | half}	interface:ethernet:csi	Sets duplex mode.
negotiation auto	interface:ethernet:csi	Configures an Ethernet port to automatically negotiate duplex mode and speed.
speed {auto | 10 | 100}	interface:ethernet:csi	Sets the speed on an Ethernet port.

B CONFIGURING EXTERNAL PROVISIONING SERVERS

A DHCP server is required for cable modems, MTAs, and CPE devices to boot and receive their IP configuration information — such as IP address and host options.

DHCP servers fall into two categories:

- **External** — DHCP servers that reside on systems other than your local Cuda 12000 (that is, the Cuda 12000 that has the cable interface that you are configuring). DHCP messages are forwarded over the network to a remote, external DHCP server. The external DHCP server can be a FastFlow Broadband Provisioning Manager (BPM) DHCP server running on another system or a third-party provisioning server running on another system.
- **Internal** — A FastFlow BPM DHCP server that resides on the same Cuda 12000 that has the cable interface you are configuring (that is, the local Cuda 12000). DHCP requests are forwarded internally to the FastFlow DHCP server. The FastFlow BPM is an optional product that may or may not be installed on your Cuda 12000.

If you are not using the internal FastFlow BPM DHCP server and are instead using an external DHCP server, then you *must* point the DHCP relay agent on the CMTS DOCSIS/EuroDOCSIS module to the IP address of the external provisioning server. The following procedure steps you through the process of configuring the CMTS to use an external DHCP provisioning server.

If a DHCP policy is not configured, then the DHCP relay drops all DHCP requests as it does not know where to forward them.

Task	Command
Note: Skip tasks 1, 2, and 3 if the FastFlow Broadband Provisioning Manager is not installed on your Cuda 12000.	
1. Enter provisioning server mode.	**prov-server**
2. Disable the Cuda 12000 integrated FastFlow Broadband Provisioning Manager server.	**ps-config serverstate disable**
3. Verify that you have disabled the provisioning server. Serverstate should be set to disabled.	**show ps-config | include ServerState** *Note that command strings are case-sensitive.*
4. Enter interface configuration mode for the selected cable interface.	**interface** <c/s/i>
5. Display the current DHCP policy configuration.	**show dhcp-policy default**
6. If the Forward Internal field in the command output from task 5. displays "enable," then disable internal DHCP forwarding. Otherwise, skip this task and proceed to the next task.	**dhcp-policy default permit forward-internal disable**
7. Specify the external DHCP server to which you want the interface to forward DHCP requests.	**dhcp-policy default permit** <ip address>

Task	Command
8. Verify the new DHCP server configuration.	**show dhcp-policy default**

Example

The following example disables the Cuda 12000 internal FastFlow Broadband Provisioning Manager DHCP server and configures the cable interface to forward DHCP requests to an external DHCP server at 192.168.23.26.

```
cli:192.168.208.3:interface:cable:csi(1/1/1)# prov-server
mode: prov-server
cli:192.168.208.3:prov-server# ps-config serverstate disable
cli:192.168.208.3:prov-server# show ps-config | include ServerState
ServerState                disabled
cli:192.168.208.3:prov-server# interface 1/1/1
mode: interface:cable:csi(1/1/1)
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-policy default
Default dhcp policy:
Policy Action                          permit
Policy Server List
Forward Internal                       enable
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-policy default permit
forward-internal disable
cli:192.168.208.3:interface:cable:csi(1/1/1)# dhcp-policy default permit
192.168.23.26
cli:192.168.208.3:interface:cable:csi(1/1/1)# show dhcp-policy default
Default dhcp policy:
Policy Action                          permit
Policy Server List         192.168.23.26
Forward Internal                      disable
```

C GLOSSARY

16 QAM

Modulation mode used by the CMTS. QAM uses both amplitude and phase modulation to encode multiple bits of data in one signaling element, thus achieving higher data transfer rates than just amplitude or phase modulation alone.

16 QAM encodes four bits per symbol as one of sixteen possible amplitude and phase combinations. 16 QAM refers to the number of discrete phase/amplitude states that are used to represent data bits.

64 QAM

A modulation mode used by the CMTS. 64 QAM uses both amplitude and phase modulation to encode multiple bits of data in one signaling element. 64 QAM encodes 6 bits per symbol as one of 64 possible amplitude and phase combinations.

256 QAM

A modulation mode used by the CMTS. 256 QAM uses both amplitude and phase modulation to encode multiple bits of data in one signaling element. 64 QAM encodes 8 bits per symbol as one of 256 possible amplitude and phase combinations.

A

Record that contains the IP address of the record's owner. Since hosts may have multiple IP addresses, multiple A records may match a given domain name.

Address Resolution Protocol (ARP)

A protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. For example, in IP Version 4, the most common level of IP in use today, an address is 32 bits long. In an Ethernet local area network, however, addresses for attached devices are 48 bits long. (The physical machine address is also known as a Media Access Control or MAC address address.)

American National Standards Institute (ANSI)	The primary organization for fostering the development of technology standards in the United States.
ARP	See Address Resolution Protocol.
Bandwidth Allocation Map	The downstream MAC Management Message that the CMTS uses to allocate transmission opportunities to CMs.
Baseline Privacy Interface	Provides data privacy for DOCSIS 1.0 CMs and CMTS. BPI+, provides privacy for DOCSIS 1.1 CMs and CMTS.
BDU	See Bridge Protocol Data Unit.
Bootstrap Protocol (BOOTP)	A protocol that lets a network user be automatically configured (receive an IP address) and have an operating system boot or initiated without user involvement. The BOOTP server, managed by a network administrator, automatically assigns the IP address from a pool of addresses for a certain duration of time.
BPI	See Baseline Privacy Interface.
Bridge Protocol Data Unit (BDU)	Spanning tree protocol messages as defined in [ISO/IEC10038].
Broadband	Network technology that multiplexes multiple, independent network carriers onto a single cable or fiber. The technology is used to carry voice, video, and data over the same cable or fiber.
Broadcast	Transmission to two or more devices at the same time, such as over a bus-type local network or by satellite; protocol mechanism that supports group and universal addressing.
Broadcast Addresses	A predefined destination address that denotes the set of all data network service access points.
Cable Modem (CM)	A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.
Cable Modem Termination System (CMTS)	A device located at the cable system head-end or distribution hub, that interfaces the HFC network to local or remote IP networks.

Cable Modem Termination System - Network Side Interface (CMTS-NSI)	The interface, defined in [DOCSIS3], between a CMTS and the equipment on its network side.
Cable Modem to CPE Interface (CMCI)	The interface, defined in [DOCSIS4], between a CM and CPE.
Carrier Hum Modulation	The peak-to-peak magnitude of the amplitude distortion relative to the RF carrier signal level due to the fundamental and low-order harmonics of the power-supply frequency.
Carrier-to-Noise Ratio (C/N or CNR)	The voltage difference between the digitally-modulated RF carrier and the continuous random noise. CNR is measured in decibels (dB).
CM	See Cable Modem.
CMCI	See Cable Modem to CPE Interface.
CMTS	See Cable Modem Termination System.
C/N or CNR	See Carrier-to-Noise Ratio.
CNAME	A record that contains an alias or nickname for the official domain name (also known as the canonical name).
Cross-Modulation	A form of television signal distortion where modulation from one or more television channels is imposed on another channel or channels.
Customer Premises Equipment (CPE)	Equipment at the end user's premises. This equipment may be provided by the end user or the service provider.
Data Link Layer	Layer 2 in the Open System Interconnection (OSI) architecture; the layer that provides services to transfer data over the transmission link between open systems.
DHCP	See Dynamic Host Configuration Protocol.
Distribution Hub	A location in a cable television network which performs the functions of a head-end for customers in its immediate area, and which receives some or all

	of its television program material from a Master Head-end in the same metropolitan or regional area.
DNS	See Domain Name System.
DOCSIS	Data Over Cable Service Interface Specification, developed by CableLabs. Defines interface standards for cable modems transmission and supporting equipment.
Domain Name System (DNS)	An on-line, distributed database used to map human-readable machine names into IP address for resolving machine names to IP addresses.
Downstream	The direction of data flow from the head-end (CMTS) to the subscriber (CM).
Drop Cable	Coaxial cable that connects to a residence or service location from a directional coupler (tap) on the nearest coaxial feeder cable.
Dynamic Host Configuration Protocol (DHCP)	A protocol that allows dynamic assignment of IP addresses to CPEs. DHCP is also used to assign IP addresses to CMs.
Dynamic Range	The ratio between the greatest signal power that can be transmitted over a multichannel analog transmission system without exceeding distortion or other performance limits, and the least signal power that can be utilized without exceeding noise, error rate or other performance limits.
Ethernet	A networking standard running speeds of 1 Gbps (Gigabit Ethernet), 10 Mbps (10BaseT) or 100 Mbps (100BaseT). Ethernet typically uses twisted pair wiring or optical fiber.
EuroDOCSIS	European Data Over Cable Service Interface Specification, developed by tComLabs and CableLabs. Defines interface standards for cable modems transmission and supporting equipment.
Extended Subsplit	A frequency division scheme that allows bidirectional traffic on a single coaxial cable. In the U.S., reverse path signals come to the head-end from 5 to 42 MHz. Forward path signals go from the head-end from 50 or 54 MHz to the upper frequency limit.
FDDI	See Fiber Distributed Data Interface.
FEC	See Forward Error Correction.

Feeder Cable	Coaxial cables that run along streets within the served area and connect between the individual taps which serve the customer drops.
Fiber Node	The interface between a fiber trunk and the coaxial distribution. Fiber nodes are located in a subscribers neighborhood.
File Transfer Protocol (FTP)	A protocol that allows users to log into a remote system, identify themselves, list remote directories, and copy files to and from the remote machine. FTP understands a few basic file formats. It is more complex than Telnet in that it maintains separate TCP connections for control and data transfer.
Flow	A unidirectional data path between a cable modem and a CMTS.
Forward Error Correction (FEC)	A technique for correcting errors incurred in transmission over a communications channel by the receiver, without requiring the retransmission of any information by the transmitter; typically it involves a convolution of the transmitted bits and the appending of extra bits, using a common algorithm by both the receiver and transmitter.
FTP	See File Transfer Protocol.
Gateway	A device that communicates with two protocols and translates services between them.
Graphical User Interface (GUI)	A program that displays information using graphics instead of command line text. The user can interact with a computer operating system through a series of “windows”, also known as “point and click”
Group Delay	The difference in transmission time between the highest and lowest of several frequencies through a device, circuit or system.
Guard Time	Minimum time allocated between bursts in the upstream referenced from the symbol center of the last symbol of a burst to the symbol center of the first symbol of the following burst. The guard time should be at least the duration of five symbols plus the maximum system timing error.
GUI	See Graphical User Interface.
Harmonic Related Carrier (HRC)	A method of spacing television channels on a cable television system in exact 6-MHz increments, with all carrier frequencies harmonically related to a common reference.

Head-End The central location on the cable network that originates the broadcast video and other signals in the downstream direction. See also Master Head-end, Distribution Hub.

Header Protocol control information located at the beginning of a protocol data unit.

HF See High Frequency.

HFC See Hybrid Fiber/Coaxial.

High Frequency (HF) The entire subsplit (5-30 MHz) and extended subsplit (5-42 MHz) band used in reverse channel communications over the cable television network.

High Return A frequency division scheme that allows bi-directional traffic on a single coaxial cable. Reverse channel signals propagate to the head-end above the downstream passband.

HRC See Harmonic Related Carrier.

Hum Modulation Undesired modulation of the television visual carrier by the fundamental or low-order harmonics of the power supply frequency, or other low-frequency disturbances.

Hybrid Fiber/Coaxial (HFC) System A broadband bidirectional shared-media transmission system using fiber trunks between the head-end and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.

Hybrid Fiber/Coaxial (HFC) Network A network where the trunk of the cable plant is fiber technology. The fiber is connected to a coaxial cable and the signal is converted so that it is compatible to that media. The coaxial cable runs through the branches of the network and is dropped into the subscriber's home.

ICMP See Internet Control Message Protocol.

IEEE See Institute of Electrical and Electronic Engineers.

IETF See Internet Engineering Task Force.

IGMP See Internet Group Management Protocol.

Impulse Noise Noise characterized by non-overlapping transient disturbances.

Incremental Related Carriers (IRC)	A method of spacing NTSC television channels on a cable television system in which all channels except 5 and 6 correspond to the standard channel plan, used to reduce composite triple beat distortions.
Information Element	The fields that make up a MAP and define individual grants, deferred grants, etc.
Ingress Noise	A type of noise that is the major source of cable system noise. It is caused by discrete frequencies picked up by the cable plant from marine and radio broadcasts or from improperly grounded or shielded home appliances such as a hair dryer.
Initial Ranging	A process in which a cable modem acquires the correct timing offset so that it can accurately transmit using the correct mini-slot. Each cable modem obtains a timing offset; the timing offset depends on the time difference of the distance of the cable modem from the CMTS. Initial ranging is performed at cable modem initialization.
Institute of Electrical and Electronic Engineers (IEEE)	An organization of electrical engineers. The IEEE fosters the development of standards that often become national and international standards. Many IEEE standards are network interface standards.
International Organization for Standardization (ISO)	An international standards body, commonly known as the International Standards Organization.
International Telecommunications Union (ITU-T)	The Telecommunication Standardization Sector of the International Telecommunications Union is the primary international body for fostering cooperative standards for telecommunications equipment and systems.
Internet Control Message Protocol (ICMP)	An Internet network-layer protocol.
Internet Engineering Task Force (IETF)	A group that defines standard Internet operating protocol, such as TCP/IP.

Internet Group Management Protocol (IGMP) A network-layer protocol for managing multicast groups on the Internet. IGMP establishes and maintains a database of group multicast addresses and the interfaces to which a multicast router must forward the multicast data packets.

Internet Protocol (IP) The method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet. An IP address includes a network address and a host address. An IP address can also be divided by using classless addressing or subnetting.

Interval Usage Code A field in MAPs and UCDs to link burst profiles to grants.

IP See Internet Protocol.

IP Filtering IP filtering enables you to filter upstream packets that pass through the CMTS. IP filtering can prevent subscribers from accessing head-end servers, enforce subscribers to log on to the cable network, enforce separately-billed service packages for data, and provide group access control for IP Multicast.

IP Multicast IP Multicast reduces traffic on a network by delivering a single stream of information to multiple users at one time.

IP Network A group of IP routers that route IP datagrams. These routers are sometimes referred to as Internet gateways. Users access the IP network from a host. Each network in the Internet includes some combination of hosts and IP routers.

IRC See Incremental Related Carriers.

ISO See International Organization for Standardization.

ITU-T See International Telecommunications Union.

Java A high level programming language developed by Sun Microsystems.

LAN See Local Area Network.

Latency The time delay, expressed in quantity of symbols, taken for a signal element to pass through a device.

Layer	A subdivision of the Open System Interconnection (OSI) architecture, constituted by subsystems of the same rank.
LDAP	See Lightweight Directory Access Protocol.
Lightweight Directory Access Protocol (LDAP)	A set of protocols for accessing information directories. LDAP is based on the standards contained within the X.500 standard, but is significantly simpler. And unlike X.500, LDAP supports TCP/IP, which is necessary for any type of Internet access to a directory server.
LLC	See Logical Link Control Procedure.
Local Area Network (LAN)	A non-public data network in which serial transmission is used for direct data communication among data stations located on the user's premises.
Logical Link Control (LLC) Procedure	In a local area network (LAN) or a Metropolitan Area Network (MAN), that part of the protocol that governs the assembling of data link layer frames and their exchange between data stations, independent of how the transmission medium is shared.
MAC	See Media Access Control.
Management Information Base (MIB)	A logical structure, used by the SNMP manager and agent, of the parameters needed for configuring, monitoring, or testing an SNMP device. The MIB is a hierarchical-naming structure used to uniquely identify SNMP objects (parameters). It is typically illustrated as an inverted tree.
Master Head-End	A head-end that collects television program material from various sources by satellite, microwave, fiber, and other means and distributes this material to Distribution Hubs in the same metropolitan or regional area. A Master Head-end MAY also perform the functions of a Distribution Hub for customers in its own immediate area.
Media Access Control (MAC) Address	A MAC address is used by the link layer protocol to forward packets "one hop at a time" between the host and the first router and between the first router and the next router and so on through the network until the packet arrives at it's final destination.
Media Access Control (MAC) Procedure	In a subnetwork, that part of the protocol that governs access to the transmission medium independent of the physical characteristics of the medium, but taking into account the topological aspects of the subnetworks, in order to enable the exchange of data between nodes. MAC

	procedures include framing, error protection, and acquiring the right to use the underlying transmission medium.
Media Access Control (MAC) Sublayer	The part of the data link layer that supports topology-dependent functions and uses the services of the Physical Layer to provide services to the logical link control (LLC) sublayer.
MIB	See Management Information Base.
Micro-reflections	Echoes in the forward transmission path due to departures from ideal amplitude and phase characteristics.
Mid Split	A frequency division scheme that allows bi-directional traffic on a single coaxial cable. Reverse channel signals propagate to the head-end from 5 to 108 MHz. Forward path signals go from the head-end from 162 MHz to the upper frequency limit. The diplex crossover band is located from 108 to 162 MHz.
Mini-Slot	A power-of-two multiple of 6.25 microsecond increments. For example, 1, 2, 4, 8, 16, 21, 64 or 128 times 6.25 microseconds. Mini-slots are used to divide the upstream bandwidth into discrete increments.
Moving Picture Experts Group (MPEG)	A group which develops standards for digital compressed moving pictures and associated audio.
MPEG	See Moving Picture Experts Group.
MSO	Multi System Operator
Multimedia Terminal Adapter (MTA)	A hardware interface between a computer and an Integrated Services Digital Network line needed for Voice Over IP.
Multipoint Access	User access in which more than one terminal equipment is supported by a single network termination.
Multipoint Connection	A connection among more than two data network terminations.

National Cable Television Association (NCTA)	A voluntary association of cable television operators which, among other things, provides guidance on measurements and objectives for cable television systems in the United States.
National Television Systems Committee (NTSC)	A committee which developed a set of standard protocol for television broadcast transmission and reception in the United States.
NCTA	See National Cable Television Association.
NEBS	See Network Equipment Building Systems.
Network Equipment Building Systems (NEBS)	NEBS is a Telcordia standard defining the physical, electrical, and environmental conditions under which network equipment must operate. NEBS includes: temperature, humidity, airborne contamination, fire resistance, earthquake and vibration, noise, electrical safety, lightning and surge immunity, ESD immunity, and electro-magnetic compatibility.
Network Layer	Layer 3 in the Open System Interconnection (OSI) architecture; the layer that establishes a path between open systems.
NS	Record that contains the domain name of the authoritative name server for the domain.
NTSC	See National Television Systems Committee.
Open Systems Interconnection (OSI)	A framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a service to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.
Open Shortest Path First (OSPF)	An Interior Gateway Routing Protocol that use link-state algorithms to send routing information to all nodes in an OSPF area by calculating the shortest path to each node based on a map of the network constructed by each node. Each router sends that portion of the routing table (keeps track of routes to particular network destinations) that describes the state of its own links, and it also sends the complete routing structure of the network.
OSI	See Open Systems Interconnection.

OSPF	See Open Shortest Path First.
Packet Identifier (PID)	A unique integer value used to identify elementary streams of a program in a single- or multi-program MPEG-2 stream.
PHY	See Physical Layer.
Physical (PHY) Layer	Layer 1 in the Open System Interconnection (OSI) architecture. It provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical, and handshaking procedures.
Physical Media Dependent (PMD) Sublayer	A sublayer of the Physical Layer that transmits bits or groups of bits over particular types of transmission link between open systems. It entails electrical, mechanical, and handshaking procedures.
PID	See Packet Identifier.
PMD	See Physical Media Dependent Sublayer.
Protocol	A set of rules and formats that determines the communication behavior of layer entities in the performance of the layer functions. It may still require an authorization exchange with a policy module or external policy server prior to admission.
PTR	A record that contains a pointer to another part of the domain name space. This record is typically used in reverse zones.
QAM	See Quadrature Amplitude Modulation.
QoS	See Quality of Service.
Quadrature Amplitude Modulation (QAM)	A method of modulating digital signals onto a radio-frequency carrier signal involving both amplitude and phase coding. This achieves a higher data transfer rate than just amplitude or phase modulation alone.
Quality of Service	A networking term that specifies a guaranteed throughput level and end to end latency for traffic on the network.
Radio Frequency (RF)	Signals that are used by the CMTS transmitter and receiver to send data over HFC network. A radio frequency carrier is modulated to encode the digital data stream for transmission across the cable network.

Request For Comments (RFC)	A technical policy document of the IETF; these documents can be accessed on the World Wide Web at http://ds.internic.net/ds/rfcindex.html.
Return Loss	The parameter describing the attenuation of a guided wave signal (e.g., via a coaxial cable) returned to a source by a device or medium resulting from reflections of the signal generated by the source.
RF	See Radio Frequency.
RF DVT	Radio Frequency Design Verification Test.
RFC	See Request For Comments.
RIP	Routing Information Protocol.
Routing Information Protocol (RIP)	A routing protocol used for IP networks. The RIP protocol calculates the shortest distance between the source and destination address based on the lowest hop count.
Service Identifier (SID)	A mapping between the CM and the CMTS based on which bandwidth is allocated to the CM by the CMTS and by which COS is implemented. Within a MAC domain, all SIDs are unique.
SID	See Service Identifier.
Simple Network Management Protocol (SNMP)	A network management protocol used to monitor IP routers, other network devices, and the networks to which they attach.
SNAP	See Subnetwork Access Protocol.
SNMP	See Simple Network Management Protocol.
SOA	Start of Authority record. The purpose of the soa record is to inform other DNS servers how to treat information that the local server provides about the domain.
SOHO	Small Office Home Office
SSRAM	Synchronous Static RAM.

Subnet A network subdivided into networks or subnets. When subnetting is used, the host portion of the IP address is divided into a subnet number and a host number. Hosts and routers identify the bits used for the network and subnet number through the use of a subnet mask.

Subnet Mask A bit mask that is logically ANDed with the destination IP address of an IP packet to determine the network address. A router routes packets using the network address.

Subnetwork Access Protocol (SNAP) An extension of the LLC header to accommodate the use of 802-type networks as IP networks.

Subscriber A user in the home who accesses a data service.

Subsplit A frequency-division scheme that allows bi-directional traffic on a single cable. Reverse path signals come to the head-end from 5 to 30 (up to 42 on Extended Subsplit systems) MHz. Forward path signals go from the head-end from 50 or 54 MHz to the upper frequency limit of the cable network.

TCP See Transmission Control Protocol.

TFTP See Trivial File-Transfer Protocol.

Tick 6.25-microsecond time intervals that are the reference for upstream mini-slot definition and upstream transmission times.

Tilt Maximum difference in transmission gain of a cable television system over a given bandwidth (typically the entire forward operating frequency range).instant at which the last bit of the same PDU crosses a second designated boundary.

TLV See Type/Length/Value.

Transmission Control Protocol (TCP) A reliable stream service which operates at the transport-layer Internet protocol which ensures successful end-to-end delivery of data packets without error.

Transmission Convergence Sublayer A sublayer of the Physical Layer that provides an interface between the Data Link Layer and the PMD Sublayer.

Transmission Medium	The material on which information signals may be carried; e.g., optical fiber, coaxial cable, and twisted-wirepairs.
Transport Stream	In MPEG-2, a packet-based method of multiplexing one or more digital video and audio streams having one or more independent time bases into a single stream.
Trivial File-Transfer Protocol (TFTP)	An Internet protocol for transferring files without the requirement for user names and passwords that is typically used for automatic downloads of data and software.
Trunk Cable	Cables that carry the signal from the head-end to groups of subscribers. The cables can be either coaxial or fiber depending on the design of the system.
Type/Length/Value (TLV)	An encoding of three fields, in which the first field indicates the type of element, the second the length of the element, and the third field the value.
UCD	See Upstream Channel Descriptor.
UDP	See User Datagram Protocol.
UHF	See, Ultra-High Frequency.
Ultra-High Frequency	The range of the radio spectrum is the band extending from 300 MHz to 3 GHz. The wavelengths corresponding to these limit frequencies are 1 meter and 10 centimeters.
Upstream	The direction of the data flow from the subscriber location (CM) toward the head-end (CMTS).
Upstream Channel Descriptor (UCD)	A MAC management message transmitted by the CMTS Adapter Module at a configured period of time. A UCD defines the characteristics of an upstream channel including the size of the mini-slot, the upstream channel ID, and the downstream channel ID. It also defines channel parameters and a burst descriptor. UCDs are transmitted for each upstream channel.
User Datagram Protocol (UDP)	In conjunction with IP, UDP provides unreliable connection-less datagram delivery service. UDP can address specific protocol ports as a destination within a given host.

Very High Frequency (VHF) The range of the radio spectrum is the band extending from 30 MHz to 300 MHz. The wavelengths corresponding to these limit frequencies are 10 meters and 1 meter.

VGA Video Graphics Array display system.

VHF See Very High Frequency.

INDEX

D

E

O

P

Q

R

S

T

U