ARSVD00729

Cornerstone

CMTS 1500 User Guide

Release 4.3   Standard 1.0   September 2003

ARRIS™

Cornerstone

CMTS 1500 User Guide

Document number: ARSVD00729
Document release: Release 4.3 Standard 1.0
Date: September 2003

Printed in the USA

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Arris Interactive.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license. A summary of the Software License is included in this document.

Publication history

September 2003

Release 4.3 Standard 1.0 version of this document.

May 2002

Release 4.2 Standard 1.0 version of this document.

March 2001

Release 4 Standard 1.0 version of this document.

Contents

Using ingress avoidance to improve system reliability 7-1

Using traps, notifications, and events to monitor system performance 8-1

Class of Service (CoS) and Quality of Service (QoS) 9-1

About this document

This document provides user operations and functions for the Cornerstone® Cable Modem Termination System 1500 (CMTS 1500), Version 4.2 software.

Audience

This document is for technicians and systems administrators who work with the Cornerstone Cable Modem Termination System (CMTS) 1500, Version 4.2 software.

CMTS 1500 documents

The documentation suite for the CMTS 1500 includes the following:

- *Cornerstone CMTS 1500 v 4.2 Installation Guide*, ARSVD00379
- *Cornerstone CMTS 1500 v 4.2 User Guide*, ARSVD00380
- *Cornerstone CMTS 1500 v 4.2 Error Codes Guide*, ARSVD00541
- *Cornerstone CMTS 1500 v 4.2 Command Line Reference Guide*, ARSVD00381
- *Cornerstone CMTS 1500 v 4.2 Release Notes*, ARSVD00382
- *Cornerstone CMTS 1500 v 4.2 Customer Letter*, ARSVD00383

Related documents

Other documents related to the CMTS 1500 include:

- *Modular Redundant Chassis Installation and Operations Guide*, ARSVD00230
- *Using the DOCSIS LCn Provisioning Server*, ARSVD00081
- *Release Notes for the DOCSIS LCn Provisioning Server*, 304764
- *Cornerstone Cable Provisioning System 2000 v1.1.1 System Administrator's Guide*, 309932-A Rev 01
- *Cornerstone Cable Provisioning System 2000 v1.1.1 User's Guide*, 309931-A Rev 01
- *Cornerstone Cable Provisioning System 2000 v1.1.1 Installation Guide*, 309928-A Rev 01
- *Cornerstone Cable Provisioning System 2000 v1.1.1 Troubleshooting Guide*, 309930-A Rev 01
- *Cornerstone Cable Provisioning System 2000 v1.1.1 Release Notes*, 309929-A Rev 01

Getting help

Documentation, customer service, and support are available from your account representative.

About the CMTS 1500

The CMTS 1500 provides broadband (cable) companies with a DOCSIS/EuroDOCSIS-based, scalable, and interoperable headend cable access solution. New features support high-speed services such as QoS (Quality of Service) and IP telephony.

New features supported by version 4.2 software on the CMTS 1500 hardware platform include:

- Redundant Ethernet port with auto-switchover
- Redundant eighth upstream receiver
- Full ingress avoidance via spectrum analyzer
- SNMP v3 support/Co-existence
- Loss of communications alarms
- BPI+ support
- Enhanced Forwarding Database (FDB)
- Provisionable age-out interval for CMs that have range but not registered
- Provisionable age-out interval for CMs that have de-registered
- Mode B Forwarding
- Mechanism for preventing Address Resolution Protocol (ARP) spoofing
- Funnel Mode
- Support for DOCSIS 1.1 features
- Support for C3M (Customer Controlled Cable Modem), including Subscriber Management and the Upstream Transmitter Disable function.

Upgrading your CMTS 1000

The CMTS 1000 can be upgraded to a CMTS 1100, which is functionally equivalent to a CMTS 1500. The upgraded CMTS 1100 will support version 4.x software Information about upgrading is available from your account representative.

When this document refers to “CMTS 1500,” the information applies to the CMTS 1100 as well.

Getting started

After performing the procedures in this chapter, the CMTS should be:

- operational on the LAN and HFC cable (RF) networks
- ready to accept connections from provisioned cable modems

These procedures in this chapter are the minimum required to begin service.

In this chapter

This chapter contains the procedures listed in Table 1-1.

Table 1-1
Procedures in this chapter

Procedure	Title	Page
1-1	Commissioning the CMTS 1500	1-3
1-2	Using the front panel	1-8
1-3	Logging into the CMTS for the first time	1-13
1-4	Pre-provisioning cable modems	1-15

Assumptions

Procedures in this chapter make the following assumptions:

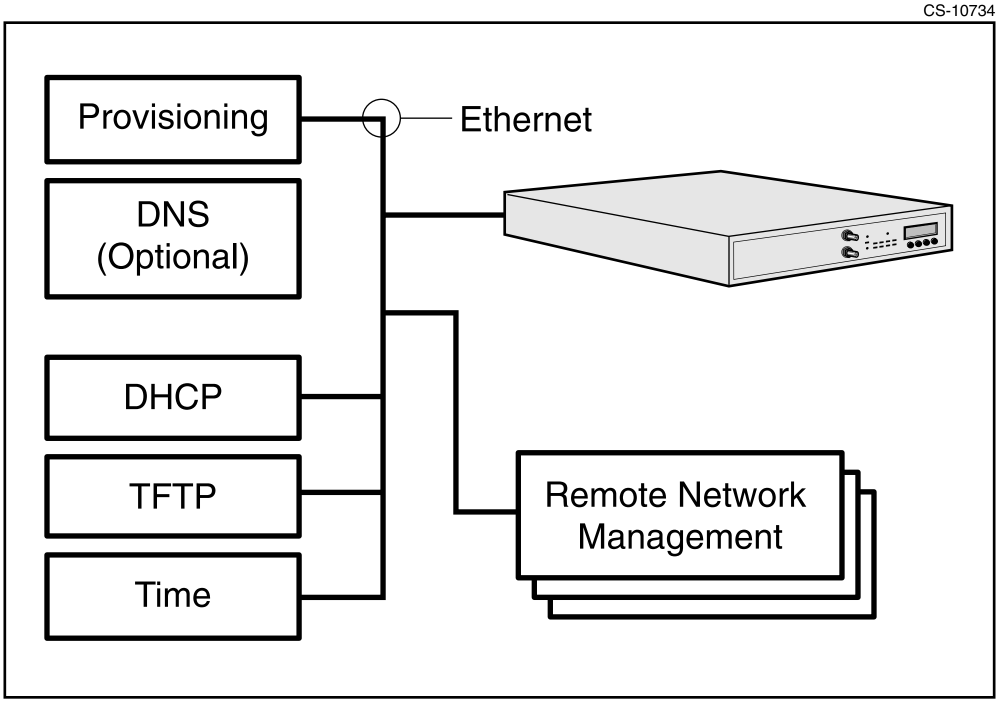
- the CMTS has been installed and powered on, according to the *CMTS 1500 Installation Guide*, ARSVD00379
- you are configuring only data support
- you want to auto-provision cable modems
- you are using the Cornerstone CPS 2000 provisioning server
- you will take as much advantage as possible of pre-provisioning
- references to cable modems include both cable modems and Packet Ports

Requirements

The procedures in this chapter require the following:

- All servers must be installed and operating properly, including the provisioning, TFTP, DHCP, DNS (optional), and time servers. The DHCP server must be set up to deliver the correct IP addresses and network masks for the CMTS and cable modem services. Figure 1-1 illustrates the interface connections between the CMTS 1500 and the various servers and network elements.
- Upstream and downstream spectrum has been allocated to cable data services.
- The HFC plant meets the following specifications:
  - — carrier-to-noise ratio (CNR) is 25 dB or better
  - — input power levels to the CMTS must be set to the proper range for the upstream modulation type and data rate you are using (the range -4 dBmV to +14 dBmV is valid for any modulation type and data rate).

Figure 1-1
Server connections



Procedure 1-1
Commissioning the CMTS 1500

Use this procedure to bring the CMTS to a minimally operational state.

Requirements

Table 1-2 lists items and information to complete this procedure.

Table 1-2
Required information

Item	Where to find it
CMTS IP address, IP subnet, mask, and domain name	your network administrator
the IP address of the gateway (router)	your network administrator
assigned downstream and upstream frequencies	cable plant engineering
TFTP server and file name to use	your account manager or network administrator
a DHCP profile assigned to the CMTS	your account manager or network administrator

If you use a provisioning server to commission the CMTS, you can perform this procedure before installing the CMTS. If you use the CLI to commission the CMTS, you must install the CMTS before proceeding. See the *CMTS 1500 Installation Guide*, ARSVD00379, for instructions.

Action

Choose the task that corresponds to the method you are using and proceed to that task.

Task	Page
Commissioning the CMTS using CPS 2000 (version 1.1)	1-4
Commissioning the CMTS using the CLI (if DHCP/TFTP not used)	1-5

—continued—

Procedure 1-1 (continued)
Commissioning the CMTS 1500

Commissioning the CMTS using CPS 2000 (version 1.1)

1 Create a new CMTS profile as follows:

- Expand the **Cable Modem Provisioning** item.
- Right-click on **CMTS Profiles**.
- Select **New CMTS Profiles** from the drop-down menu.

2 Enter the following information in the CMTS profile:

- Downstream frequency
- Upstream frequencies
- TFTP file name to use
- TFTP server address
- proper cable standard for your locale (DOCSIS or EuroDOCSIS)

3 Access the CMTS configuration area in the provisioning server as follows:

- Expand the **Cable Modem Provisioning** item.
- Expand the **Cable Plant** item.
- Right-click on **Regional Headend**.
- Choose **New CMTS** from the drop-down menu.

The New CMTS dialog appears.

4 Enter the following information in the CMTS configuration area:

- CMTS IP address
- subnet mask
- domain name
- CMTS Ethernet MAC address (from the label on the back of the CMTS)
- CMTS profile (created in steps 1 and 2)

5 Apply the DHCP profile for the CMTS as follows:

- Click the **DHCP Options** tab in the New CMTS dialog.
- Select the **Apply Template** button.
- Choose the proper template from the dialog.

—continued—

Procedure 1-1 (continued)
Commissioning the CMTS 1500

Commissioning the CMTS using the CLI (if DHCP/TFTP not used)

1 Log into the CMTS using the console port. See “Logging into the CMTS for the first time” on page 1-13 for details.

2 Configure basic CMTS IP parameters using the following commands:

```
[] Console> manage ↵
[] box# ip-level ↵
[] ip-level# info ↵
[] ip-level# config-ip-address <cmtsip> ↵
[] ip-level# config-ip-subnet <subnet> ↵
[] ip-level# config-ip-gateway <gwip> ↵
[] ip-level# dns-control enabled ↵
[] ip-level# back ↵
[] box# admin ↵
[] admin# provisioning-control use-nvram ↵
```

where	is the…
<cmtsip>	IP address assigned to the CMTS
<subnet>	IP subnet mask associated with the CMTS IP address
<gwip>	IP address of the gateway on this network (typically a router)

Note: To implement the CMTS IP parameters listed above, you must run the *admin* command and use NVRAM.

3 Type **exit** to return to the Console> prompt.

—continued—

Procedure 1-1 (continued)
Commissioning the CMTS 1500

4 Configure the networking information using the following required commands:

```
[] box# admin↵
[] admin# config-tftp-ip-addr <tftpip> ↵
[] admin# config-tftp-filename <tftpfile> ↵
[] admin# sw-server-ip-addr <swip> ↵
[] admin# sw-filename <swfile> ↵
```

where	is the…
<tftpip>	IP address of the TFTP server
<tftpfile>	name of the .MD5 configuration file on the TFTP server
<swip>	IP address of the server containing the CMTS software album
<swfile>	name of the CMTS software file (for example, “CMTS_ALBUM_4.2.0”); this file must be available on the TFTP server

Note: The **sw-server-ip-addr** and **sw-filename** commands are required only if you are upgrading the CMTS software.

—continued—

Procedure 1-1 (continued)
Commissioning the CMTS 1500

5 Configure the networking information using the following optional commands:

```
[] admin# bootp-relay-control <bootp> ↵
[] admin# time-rfc868-addr <timeip> ↵
[] admin# time-offset <offset> ↵
[] admin# time-sntp-addr <sntpip> ↵
```

where	is the…
<bootp>	BOOTP relay control: • **disabled** (default) • **relay-only-enabled** • **relay-tagging-enabled**
<timeip>	IP address of the time server (optional)
<offset>	offset, in seconds, from GMT (optional)
<sntpip>	IP address of the SNTP time server (optional)

6 Configure the downstream RF interface as described in “Configuring downstream RF parameters using the CLI” on page 3-4.

7 Configure the upstream RF interfaces as described in “Configuring upstream RF parameters using the CLI” on page 3-9.

—end—

Procedure 1-2
Using the front panel

Use the tasks in this procedure to display information about the CMTS 1500 in the front panel.

Action

Choose the task that corresponds to the method you are using and proceed to that task.

Task	**Page**
Viewing system information	1-8
Viewing downstream information	1-10
Viewing upstream information	1-11
Selecting an upstream channel directed to a test port	1-12

Viewing system information

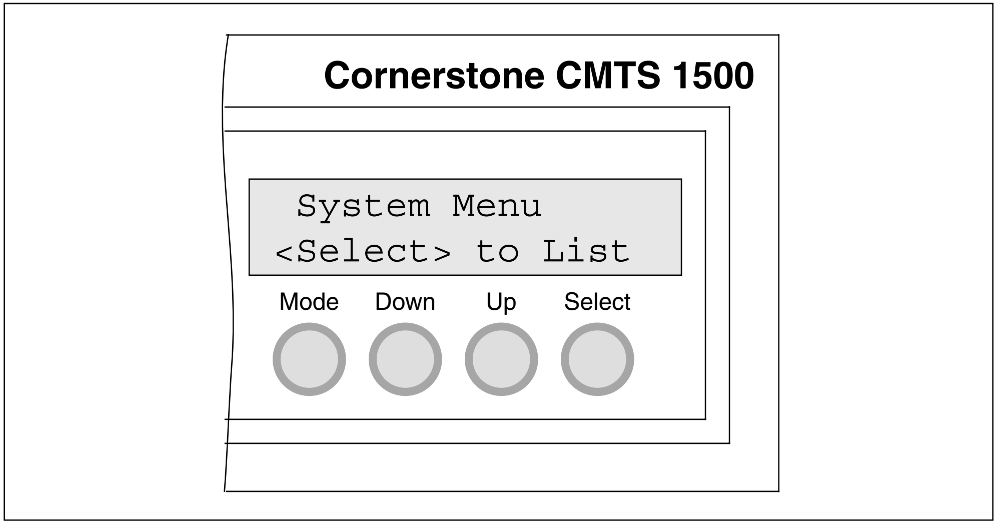
Use the following steps to view the CMTS 1500 system operating information:

1 Press **Mode** until the front panel displays “`Cornerstone`.”

2 Press **Down**.

The System menu shown in Figure 1-2 displays.

Figure 1-2
System menu



—continued—

Procedure 1-2 (continued)
Using the front panel

3 Press **Select** to display the system information.

4 Press the **Up** or **Down** buttons to navigate the system information described in Table 1-3 and select the desired parameter.

Table 1-3
CMTS 1500 system parameters

Display	Description
System Name	Displays the system name assigned to the CMTS 1500. If no name is assigned, then the second line is blank.
System State	Displays the current operational state of the CMTS 1500: • **Up** = CMTS is operational; RF is present on both upstream and downstream channels. • **Up(Alert)** = CMTS is operational; RF is present on both upstream and downstream channels; and one or more upstreams are down. • **Down** = CMTS is not transmitting RF • **Quiet** = CMTS is halted • **ReqSvc** = CMTS requires service or corrective action
System Uptime	Displays time up since last system boot cycle. Format is: Days:Hours:Minutes:Seconds
System Modems	Displays number of active modems recognized by the CMTS 1500.
System Ip Addr	Displays the IP address assigned to the CMTS 1500 (by the DHCP server). If no IP address is assigned, then the display reads **0.0.0.0**.
System MAC Addr	Displays the unique MAC address for the CMTS 1500 (which should match the MAC address on the rear panel label).
System Revision	Displays the CMTS 1500 main circuit board hardware version and the current software version loaded.
System Temp	Displays the CMTS 1500 internal temperature (in degrees Fahrenheit and in degrees Celsius).
System Fan Sts	Displays the CMTS 1500 fan status. This feature is used to detect and report problems with the front, middle, or back fans in the CMTS 1500.

—continued—

Procedure 1-2 (continued)
Using the front panel

Viewing downstream information

1 Press **Mode** until the front panel displays "`Cornerstone`."

2 Press **Down** until the front panel displays "`DnStm`."

3 Press **Select** to display the downstream information.

4 Press the **Up** or **Down** buttons to navigate the downstream information described in Table 1-4 and select the desired parameter.

Table 1-4
CMTS 1500 downstream parameters

Display	**Description**
DnStm State	Displays the current transmitting state, power, and modulation type for the Downstream channel. The transmit state is one of the following: • UP = CMTS 1500 is transmitting. • DOWN = CMTS 1500 is not transmitting. The modulation pattern is one of the following: • Q64 = QAM 64 • Q256 = QAM 256 • RevSvc = used for maintenance only.
DnStm Cent Freq	Displays the channel plan and the center frequency (in MHz) for the downstream channel. The channel plan is one of the following: • STD = Standard • EURO = EuroDOCSIS Plan • HRC = Harmonically Related Carrier • IRC = Incrementally Related Carrier
DnStm Chan/Plan	Displays the Channel Number (Chan) and the Frequency Plan (FP) for the Downstream channel.
DnStm Power Lvl	Displays the downstream power level (measured at the CMTS 1500 Downstream port) in dBmV. This value is accurate within +/- 1 dBmV.

—continued—

Procedure 1-2 (continued)
Using the front panel

Viewing upstream information

1 Press **Down** until the front panel displays "`UpStm 1 Menu`." This is the entry for the first upstream receiver.

2 Continue to press **Down** until the front panel displays the upstream receiver that you want to view.

3 Press **Select** to display the upstream information.

4 Press the **Up** or **Down** buttons to navigate the upstream information described in Table 1-5 and select the desired parameter.

Table 1-5
CMTS 1500 upstream parameters

Display	Description
UpStm1 State	Displays the channel port and its current state. • UP = Upstream port is receiving. • DOWN = Upstream port is not receiving.
UpStm Cent Freq	Displays the upstream channel port and its center frequency (in MHz).
UpStm1 ModFormat	Displays the channel port, and its modulation format (QPSK or QAM 16).
UpStm1 Bandwidth	Displays the upstream channel port and its bandwidth (in kHz).
UpStm1 Input Lvl	Displays the channel port and its input level in dBmV.
UpStm1 Test Port	Redirects the selected upstream to the front panel test port (press **Select** to redirect the upstream).

5 Repeat steps 2 through 5 to view remaining upstream channels.

Note: After selecting an upstream channel using the front panel display, there is no indication on the front panel that the action has occurred. You cannot reverse the action by pressing the select button a second time. The UsTestPort parameter in the lcCmtsIfInfo MIB object under the LanCity MIB indicates the action has occurred.

—continued—

Procedure 1-2 (continued)
Using the front panel

Selecting an upstream channel directed to a test port

1 Press the **Up** and **Down** buttons to select the "`UpStm 1 Menu`".

2 Press the **Up** and **Down** buttons to select the upstream test port option. The port should be displayed as "not-in-use".

3 Press the **Select** button. The port should now show the upstream as re-directed to the test port.

4 Press the **Select** button again to stop re-direction to the test port.

Procedure 1-3
Logging into the CMTS for the first time

The CMTS has a serial port connection on the rear panel (marked CONSOLE) which you can use to connect the CMTS to a host computer, a display terminal, or a modem. Typically, this Console port connection is used to control the CMTS through the Command Line Interface (CLI). See the *CMTS 1500 CLI Reference Guide*, ARSVD00381, for more information.

> ***Note:*** You must use the console port connection the first time you use the Command Line Interface.

Perform this procedure after installing the CMTS 1500 for the first time.

Requirements

You need the following items to perform this procedure:

- A VT-100 compatible terminal, or a PC with terminal emulation software
- A null-modem cable with a female DB-9 connector
- A small flat blade screwdriver (to secure the cable)

Action

Follow these steps to log onto the CMTS through the console port.

1 Connect the cable from the CMTS CONSOLE port to the terminal or PC. Secure the end of the cable using the screwdriver, if necessary.

2 Configure the terminal or emulation program as follows:

- Data rate = **9600**
- Data Bits = **8**
- Stop Bits = **1**
- Parity = **None**
- Flow Control = **None (Off)**

See the documentation for your terminal or emulation program for details. If you are using a PC, make sure the emulation program is set to use the port (COM1 or COM2) that you have connected the cable to.

3 Activate the connection (refer to the terminal or emulation program documentation for details) and press **Return**.

The CMTS should respond with a login: *prompt. If not, make sure the cable is of the proper type and the serial port is configured correctly. If the console port has been used before, the CMTS settings may be changed from their defaults.*

—continued—

Procedure 1-3 (continued)
Logging into the CMTS for the first time

4 Respond as follows to the prompts:

Prompt	**Response**
`login:`	type `root` and press the **Return** key
`password:`	press the **Return** key

The Console> *prompt appears.*

5 Perform the following in any order:

- Set a password for the admin account as described in “Creating and modifying user accounts using the CLI” on page 4-7.
- Create other accounts as needed, using “Creating and modifying user accounts using the CLI” on page 4-7.
- Assign an IP address to the CMTS, using the task “Commissioning the CMTS using the CLI (if DHCP/TFTP not used)” on page 1-5.

6 When you are ready to log out, type `logout` and press the **Return** key.

The CMTS disconnects you.

—end—

Procedure 1-4
Pre-provisioning cable modems

Use this procedure to configure cable modems. The cable modems need not be installed before performing this procedure.

Requirements

You need the following items and information to complete this procedure:

- a provisioning server, such as the CPS 2000
- assigned downstream and upstream frequencies
- TFTP server and file name to use
- a DHCP profile assigned to the CMTS
- the CMTS IP address, mask, and domain name

Action

Follow these steps to pre-provision cable modems. For details about each step, see the documentation that accompanies your provisioning manager.

1. Using the provisioning manager, configure IP filters for the cable modems. The following restrictions apply:
   - If you are using auto-provisioning, the filter must allow ICMP packets to the registration DNS server.
   - If you are using auto-provisioning, the filter must allow TCP packets from the registration DNS server.
   - Port 53 (TCP) must be accepted in either direction to allow DNS traffic.
2. Using the provisioning manager, create service classes as desired. For auto-provisioning, create a service class for unregistered modems as well.
3. Using the provisioning manager, create a profile for the cable modems. The following restrictions apply:
   - The upstream and downstream frequencies specified in this profile must match at least one CMTS on the network. The cable modems using this profile must be able to reach this CMTS.
   - Unregistered modems should use IP filters that redirect all traffic to the registration DNS server.
   - The specified TFTP server and file name must exist and be reachable.

—end—

Procedure 1-5
Logging into the CLI using Telnet or Secure Shell

The CMTS supports up to five simultaneous remote network logins using either *telnet* or SSH (Secure Shell).

Requirements

Perform “Logging into the CMTS for the first time” on page 1-13 at least once before using this procedure. The CMTS needs an IP address and a root password assigned before it can accept network logins.

You also need a *telnet* or SSH client program installed on your remote computer. Both *telnet* and SSH clients are available for all popular operating systems.

Action

Follow these steps to log on to the CMTS.

1 Start the *telnet* or SSH client program on your remote computer. See the documentation for your operating system and client for details.

2 Connect to the CMTS, specifying one of the following:

- The IP address—a series of four numbers separated by dots; for example, **10.1.1.250**
- The FQDN—a fully-qualified domain name assigned through your Domain Name System (DNS) server; for example, **cmts1.location.provider.net**

The CMTS displays a login: *prompt on your remote computer.*

3 Enter your account name and password as prompted.

The CMTS displays the Console> *prompt.*

4 Perform any necessary procedures.

Note: If you do not enter any commands for five minutes, the CMTS disconnects you. You can configure the CMTS to specify a different inactivity disconnect time.

5 When you are ready to log out, type `logout` and press the **Return** key.

The CMTS disconnects you.

—end—

Using auto-provisioning to simplify subscriber signup

After performing the procedures in this chapter, your cable data network will support auto-provisioning. Using auto-provisioning, subscribers can install their own cable modems and register (sign up for service) online. Until a subscriber registers, the only available Internet service is the web page that provides the registration service.

In this chapter

This chapter contains the procedures listed in Table 2-1.

Table 2-1
Procedures in this chapter

Procedure	Title	Page
2-1	Configuring the CMTS to support auto-provisioning	2-3
2-2	Configuring the registration DNS server	2-4
2-3	Configuring the registration web page	2-5
2-4	Configuring filters for unregistered cable modems2-4	2-6

Reference

The procedures in this chapter assume that you are using CPS 2000, version 1.1 or newer, to provision your network. See Chapter 7, “Auto-Provisioning,” in the *CPS 2000 User’s Guide* for details and important limitations.

Requirements

The procedures in this chapter require the following:

- The CMTS has been provisioned and installed successfully. At a minimum, perform the procedures in Chapter 1 of this manual.
- The CPS 2000 is set up for auto-provisioning as described in Chapter 7, “Auto-Provisioning,” in the *CPS 2000 User’s Guide*.
- You must have a web server, accessible from the cable data network, dedicated to handling the registration page. The server and registration software must be able to connect to your billing system to set up the subscriber’s account. Contact your next level of support for information.

Procedure 2-1
Configuring the CMTS to support auto-provisioning

This procedure sets up the CMTS to support auto-provisioning of cable modems.

Use this procedure under the following conditions:

- when setting up a CMTS for the first time
- when the CMTS has been reset to factory defaults (using the **reset factory** command)

Requirements

You must be logged into the CMTS CLI with administrative privileges.

Make sure the CMTS is displaying the Console> prompt before starting.

Action

Follow these steps to configure the CMTS to support auto-provisioning.

1 Configure the first BOOTP server by typing the following commands, starting at the Console> prompt:

```
Console> manage ↵
[]box# admin ↵
[]admin# bootp-relay-control relay-tagging-enabled ↵
[]admin# bootp-modify/1 ↵
[]bootp-modify/1# server-ip-address <dhcpaddr> ↵
[]bootp-modify/1# client-types-relayed  any ↵
[]bootp-modify/1# status active ↵
```

where	is the…
<dhcpaddr>	IP address of the DHCP server

2 Type **exit** to return to the Console> prompt.

—end—

Procedure 2-2
Configuring the registration DNS server

Use this procedure to configure the CPS 2000 to provide a registration DNS server.

> ***Note:*** Any DNS server may act as a registration DNS server. If you want to use a different DNS server, follow the instructions accompanying the server software instead of this procedure.

Action

1 If you are running a standard DNS server on the CPS 2000, you must terminate it. You cannot run the standard DNS and registration DNS servers on the same system.

2 Start the registration DNS server as described in the *Cornerstone CPS 2000 System Administrator's Guide*, 309932-A:

- For Unix systems, see Chapter 24.
- For NT systems, see Chapter 23.

3 (optional) To change the IP address of the web server hosting the subscriber registration page, do the following:

- For Unix systems, set the IPTORETURN environment variable to the IP address of the web server.
- For NT systems, set the IpToReturn registry variable to the IP address of the web server.

Note: The default IP address is that of the CPS 2000 server.

—end—

Procedure 2-3
Configuring the registration web page

The CPS 2000 includes a generic web provisioning agent, that allows a web-based application (or any other application) to send registration information to the CPS 2000. In turn, the CPS 2000 has translators that interact with the following billing systems:

- DST Innovis
- CSG Systems

Web server location

The CPS 2000 Application Server provides web services for logging into the CPS 2000 and for other applications that interact with it. The $CPS_HOME environment (shell) variable specifies the directory that contains HTML pages to be accessed through the web server. Common files stored in this location are:

- index.html—the registration web page
- CPS_login.html—the CPS 2000 login page

In addition, you need to create a CGI (Common Gateway Interface) web application that verifies subscriber input then sends it to the CPS 2000 and to your billing server. See Chapter 9 in the *Cornerstone CPS 2000 System Administrator's Guide*, 309932-A, for more information.

—end—

Procedure 2-4
Configuring filters for unregistered cable modems

Follow these steps to configure an IP filter that directs unregistered subscribers to the registration web server.

Requirements

You must be logged into the CPS 2000, with appropriate privileges, to create an IP filter.

Action

1 In the tree area (the left side of the CPS 2000 screen), expand the **Cable Modem Provisioning** root object.\

2 Under Cable Modem Provisioning, expand the **Configuration Settings** object.

3 Under Configuration Settings, click on the **IP Filters** object.

Registration and Clear filters appear in the window on the right side of the CPS 2000 screen.

4 Right-click on the **Registration** IP filter and choose **Update IP Filter** from the drop-down menu.

5 Add the six filters shown in Table 2-2. Replace <srvIP> with the IP address of the registration web server.

Table 2-2
Registration IP filter settings

Ord	Status	CbS	Control	IntID	Dir	Brdcast	SrcIP	SrcMask	DestIP	DestMask	Prot	LSprt	HSprt	LDprt	HDprt
1	C and G	Yes	Accept	Both	Both	No	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	ICMP				
2	C and G	Yes	Accept	Both	Both	No	0.0.0.0	0.0.0.0	<srvIP>	255.255.255.255	TCP				
3	C and G	Yes	Accept	Both	Both	No	<srvIP>	255.255.255.255	0.0.0.0	0.0.0.0	TCP				
4	C and G	Yes	Accept	Both	Both	Yes	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	UDP				
5	C and G	Yes	Accept	Both	Both	No	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	UDP				
6	C and G	Yes	Accept	Both	Both	No	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	TCP	53	53	53	53

6 Click **OK** to save the IP filter settings.

—end—

Provisioning the CMTS RF interfaces

This chapter provides procedures for manually configuring upstream and downstream RF (cable) parameters.

Purpose of this chapter

Use the procedures in this chapter:

- if you intend to deploy more than one upstream or downstream on a single physical path; and you are not using a provisioning server to set up the CMTS, or if you need to change RF parameters after commissioning the CMTS
- to configure the function of the upstream receiver in slot 8—this receiver can act as a standard upstream, a redundant upstream, or a spectrum analyzer (for ingress avoidance or debugging purposes)

In this chapter

This chapter contains the procedures listed in Table 3-1.

Table 3-1
Procedures in this chapter

Procedure	Title	Page
3-1	Configuring downstream RF parameters	3-4
3-2	Configuring upstream RF parameters	3-8
3-3	Configuring the eighth upstream receiver using the CLI	3-13
3-4	Configuring the eighth upstream receiver using SNMP	3-17

Terms and concepts

This chapter uses the following terms and concepts:

Upstream
The RF (and fiber) path from a cable modem to the CMTS.

Downstream
The RF (and fiber) path from the CMTS to a cable modem.

Carrier path
The physical upstream route from a group of cable modems to the CMTS.

Default parameters

The CMTS has the following factory default RF parameter settings:

- The admin status of the cable is down. Note: the downstream frequency must be set prior to bringing the admin status up.

Default downstream RF parameters

- Frequency: 0 MHz
- Transmit power levels: 51 dBmV
- Channel width: 6 MHz for DOCSIS, 8 MHz for EuroDOCSIS
- Cable interface administrative status: down
- Modulation: QAM64

Default upstream RF parameters

The following are factory default parameters for all upstreams.

- Frequency: see Table 3-2
- Receive power: 11 dBmV
- Channel width: 3.2 MHz
- Modulation profile: 1

Table 3-2
Default upstream frequencies

Upstream	Frequency
1	26.75 MHz
2	29.95 MHz
3	31.15 MHz
4	36.35 MHz
5	39.55 MHz
6	23.55 MHz
7	20.35 MHz
8	17.15 MHz

Procedure 3-1
Configuring downstream RF parameters

Use this procedure to change the downstream frequency, modulation, and transmit power.

Action

Identify the task that corresponds with your preferred interface and follow the steps in that task.

Task	Page
Configuring downstream RF parameters using the CLI	3-4
Configuring downstream RF parameters using SNMP	3-6

Configuring downstream RF parameters using the CLI

1 Configure the basic downstream channel parameters using the following commands:

[] Console> **manage** ↵

[] box# **cable-level** ↵

[] cable-level# **downstream** ↵

[] downstream# **frequency <dsfreq>** ↵

[] downstream# **width <bandwidth>** ↵

[] downstream# **modulation <modtype>** ↵

where	is the…
<dsfreq>	center frequency, in Hz, of the downstream channel (default: **0**; typical: **681000000**)
<bandwidth>	bandwidth, in Hz, of the downstream channel: • **6000000** (6 MHz) for DOCSIS (default) • **8000000** (8 MHz) for EuroDOCSIS
<modtype>	modulation scheme; one of: • **qam64** (default) • **qam256**

—continued—

Procedure 3-1 (continued)
Configuring downstream RF parameters

2 Check the signal formatting type, and change it if necessary, using the following commands:

```
[] downstream# ? ↵
[] downstream# annex <type> ↵
```

where	
<type>	is the downstream signal formatting type; one of: • **annex-b** (default) — North American DOCSIS • **annex-a** — EuroDOCSIS

3 (optional) Configure other downstream parameters, if necessary, using the following commands:

```
[] downstream# interleave <interleave> ↵
[] downstream# power <dspower> ↵
```

where	is the…
<interleave>	number of traps and increments for this port (default: **taps8increment16**)
<dspower>	power level, in 0.1 dBmV increments, of the downstream channel (default: **510**; 51 dBmV) ***Note:*** If you change the default power level, check and balance network power levels as well.

4 Check the frequency split, and change it if necessary, using the following commands:

```
[] downstream# back ↵
[] cable-level# ? ↵
[] cable-level# frequency-split ↵
[] frequency-split# info ↵
[] frequency-split# frequency-split <split> ↵
```

where	
<split>	is the frequency split setting; one of: • **standard** (default) — North American DOCSIS (5-42 MHz / 65-860 MHz) • **euro** — EuroDOCSIS (5-65 MHz / 100-862 MHz)

5 Type **exit** to return to the Console> prompt.

—continued—

Procedure 3-1 (continued)
Configuring downstream RF parameters

Configuring downstream RF parameters using SNMP

1 Create an entry in the docsIfDownstreamChannelEntry table and set the following objects for the entry:

Object	Value
docsIfDownChannelFrequency	center frequency, in Hz, of the downstream channel
docsIfDownChannelWidth	bandwidth, in Hz, of this channel
docsIfDownChannelModulation	the modulation type for this channel: • **unknown** • **other** • **qam64** • **qam256**
docsIfDownChannelInterleave	the Forward Error Correction (FEC) interleaving used for this channel: • **taps8Increment16** (default) (protection 5.9/4.1 µsec, latency .22/.15 msec) • **taps16Increment8** (protection 12/8.2 µsec, latency .48/.33 msec) • **taps32Increment4** (protection 24/16 µsec, latency .98/.68 msec) • **taps64Increment2** (protection 47/33 µsec, latency 2/1.4 msec) • **taps128Increment1** (protection 95/66 µsec, latency 4/2.8 msec) • **taps12Increment17** (EuroDOCSIS default) • **taps204Increment1** (EuroDOCSIS)
docsIfDownChannelPower	transmit power (in 0.1 dBmV increments) (default **510**; 51 dBmV)

—continued—

Procedure 3-1 (continued)

Configuring downstream RF parameters

2 Open the lcCmtsIfConfig object. Check the signal formatting type, and change it if necessary, using the following objects:

Object	**Value**
lcCmtsAnnex	the downstream signal formatting type; one of: • **annex-b** (default) — North American DOCSIS • **annex-a** — EuroDOCSIS
lcCmtsFrequencySplit	the upstream/downstream frequency split for the cable system; one of: • **standard** (default) — North American DOCSIS (5-42 MHz / 65-860 MHz) • **euro** — EuroDOCSIS (5-65 MHz / 100-862 MHz)

—end—

Procedure 3-2
Configuring upstream RF parameters

Use this procedure to change the upstream frequency, modulation, and receive power.

Action

Identify the task that corresponds with your preferred interface and follow the steps in that task.

Task	**Page**
Configuring upstream RF parameters using the CLI	3-9
Configuring upstream RF parameters using SNMP	3-10

—continued—

Procedure 3-2 (continued)
Configuring upstream RF parameters

Configuring upstream RF parameters using the CLI

1 Configure an upstream channel using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# upstream-list ↵
[] upstream-list# upstream-specific/<channel> ↵
[] upstream-specific/4# frequency <usfreq> ↵
[] upstream-specific/4# width <bandwidth> ↵
[] upstream-specific/4# power <uspower> ↵
[] upstream-specific/4# input-power-window <range> ↵
[] upstream-specific/4# modulation-profile <prof> ↵
```

where	is the…
<channel>	upstream channel you are configuring: **4** to **11** (where **4** represents the first upstream, and so on)
<usfreq>	center frequency, in Hz, of the upstream channel. ***Note:*** If two upstreams share a common physical path, they must use different frequencies.
<bandwidth>	width, in Hz, of the upstream channel; one of: **200000**, **400000**, **800000**, **1600000**, or **3200000** (default)
<uspower>	receive power at the CMTS inputs in 0.1 dBmV increments (default: **110** for 11 dBmV)
<range>	size of the receive power level range, in 0.1 dBmV increments: **20** to **150** (default: **60** for 6 dB) The power window is useful when cable modems are transmitting at or near the maximum or minimum power output level, but are unable to reach the set receive level. Adjusting the window allows cable modems to register with the CMTS.
<prof>	modulation profile index number assigned to this upstream channel (default: **1**) This should be changed only if you wish to select a different modulation profile

—continued—

Procedure 3-2 (continued)
Configuring upstream RF parameters

2 Repeat step 1 for each upstream channel you want to configure.

3 Type **exit** to return to the Console> prompt.

Configuring upstream RF parameters using SNMP

1 Select one of the eight upstream entries in the docsIfUpstreamChannelEntry table and set the following objects for the entry:

Object	**Value**
docsIfUpChannelFrequency	center frequency, in Hz, of the upstream channel (default: see Table 3-2)
docsIfUpChannelWidth	bandwidth, in Hz, of this upstream channel (default **3200000**)
docsIfUpChannelModulationProfile	the modulation profile for this upstream channel (corresponds to an entry in the docsIfCmts-ModulationEntry table; default is **1**)
docsIfUpChannelSlotSize	the number of 6.25 microsecond ticks in each upstream mini-slot; default is **4**
docsIfUpChannelRangingBackoffStart	the initial random backoff window to use when retrying Ranging Requests; default is **2**
docsIfUpChannelRangingBackoffEnd	the final random backoff window to use when retrying Ranging Requests; default is **5**
docsIfUpChannelTxBackoffStart	the initial random backoff window to use when retrying transmissions; default is **3**
docsIfUpChannelTxBackoffEnd	the final random backoff window to use when retrying transmissions; default is **10**

Note: When a CMTS 1500 is connected to different upstream carrier paths, specific configuration rules apply. (For example, when the return paths coming from multiple optical receivers are connected to their own individual upstream channels.)

Since there is only one downstream and multiple upstream channels associated with this downstream, cable modems that have not ranged and have no upstream channel ID saved into NVRAM can pick any upstream channel to perform initial ranging.

In the following example on a CMTS 1500, the upstream paths have two different returns from upstream receivers at the headend. The downstream frequency is set to 681 MHz.

Upstream Channel ID	Center	Width	Modulation
1	20 MHz	3.2 MHz	QPSK, US Path 1
2	21 MHz	3.2 MHz	QAM16, US Path 2
2	20 MHz	3.2 MHz	QAM16, US Path 3

The cable modem is physically connected to upstream path 2. When it initializes, it performs the following steps:

1 Scans the downstream, acquires QAM lock at 681 MHz, acquires FEC lock, and decodes MPEG packetization.

2 Waits for upstream channel descriptor messages and builds a list of available upstream channel identifiers.

3 Randomly picks an upstream channel ID from an available list.

4 Waits for MAPs corresponding to the upstream channel descriptor.

5 Decodes MAP messages and locates an initial maintenance interval.

6 Sends an initial ranging request when the initial maintenance interval that corresponds to the upstream channel ID arrives.

In Step 3, the cable modem may select upstream channel ID 1 descriptor and upstream path 1. In this case, the cable modem waits for UCID 1 MAPs and locates the first CMTS initial maintenance interval, scheduled for the upstream channel ID. An initial ranging request is sent when the initial maintenance interval arrives.

In this case, the initial ranging request arrives at the CMTS on upstream channel 2, because upstream channel ID 2 center frequency is 21 MHZ, and the width is 3.2 MHz. The message is sent at 20 MHz, and the width is sent at 21 MHz. Notice, there is an overlap between 19.4 MHz and 21.6 MHz.

The solution to this is to use non-overlapping upstream channel frequencies or groups of non-overlapping upstream channel frequencies on all upstream channels. (These are called channel groups.)

The modulation profile on all upstream channels, sharing the same frequency, should be identical. The following is a sample configuration:.

Upstream Channel ID	Center	Width	Modulation
1	20 MHz	3.2 MHz	QPSK, US Path 1
2	24 MHz	3.2 MHz	QAM16, US Path 2
2	20 MHz	3.2 MHz	QPSK, US Path 3

In the above configuration, upstream channel ID 1 and upstream channel ID 3 are part of a channel group. Upstream channel ID 2 is part of another channel group.

Even though the CMTS can decode the initial ranging request sent by the cable modem, it does not guarantee that the initial maintenance interval for upstream channel ID 1 is identical to the initial maintenance interval for upstream channel ID 2. The initial ranging request, sent by the cable modem, may interfere with traffic (e.g., a cable modem transmitting user data) because the two initial maintenance intervals are not synchronized.

Ensure the initial maintenance intervals are synchronized and use the "*multi-us-config/<channel {4-11}*" command to set the carrier path and channel group parameters for the specified upstream channel, located under "*manage/cable-level*". This command must be set via the CLI. Refer to the *CMTS 1500 CLI Reference Guide*, ARSVD00123 for additional information on this command.

Note that all upstream channels sharing the same frequency configuration and modulation profile must be part of the same channel group.

—end—

Procedure 3-3 **Configuring the eighth upstream receiver using the CLI**

The Cornerstone CMTS provides special functionality for an upstream receiver module installed in slot 8. You can configure the receiver to operate in any of the modes listed in Table 3-3.

Table 3-3
Upstream receiver modes

Mode	Description	Benefits	Considerations
standard	The eighth upstream receiver acts as a standard upstream.	All eight upstream receivers can carry subscriber traffic.	Protection and spectrum analysis is not available.
redundant (failover)	The eighth upstream receiver operates in a standby mode; it can automatically or manually take over for a failed upstream.	More robust subscriber service.	Spectrum analysis not available.
spectrum analyzer	The eighth upstream receiver constantly scans selected upstream frequencies and records noise data.	Ingress avoidance frequency hops always transfer to a frequency with low noise	Operator control is not available when ingress avoidance is enabled.
redundant and spectrum analyzer	The eighth upstream receiver acts as a spectrum analyzer unless a failover switch is necessary.	• More robust subscriber service • The spectrum analyzer can be used unless the redundant receiver is switched in	A protection switch disables spectrum analysis.
debug spectrum analyzer	The eighth upstream scans selected upstream frequencies under operator control, and records noise data.	Operators can scan specific parts of the spectrum to look for problems	No other functions are available while the debug spectrum analyzer is in use.

You can configure the eighth upstream receiver as both a redundant upstream and a spectrum analyzer, with the considerations shown in Table 3-3.

—continued—

Note: In order for a failed upstream to automatically failover to the eighth upstream, you must install upstream receivers in slots 1, 8, and in at least one other slot. Furthermore, you must have at least 20 modems attached to upstream 1, and attach at least one modem to the other installed upstream receivers.

All RF switching is done internally in the CMTS. Therefore, you do not need a coax cable to connect to the eighth port.

Action

Perform the following tasks in any order when needed.

Task	Page
Configuring the receiver function	3-15
Manually switching the upstream to the redundant receiver	3-16

—continued—

Procedure 3-3 (continued)
Configuring the eighth upstream receiver using the CLI

Configuring the receiver function

1 Configure the eighth upstream receiver using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ? ↵
[] cable-level# multi-usage-us ↵
[] multi-usage-us# admin <type> ↵
```

where	is the…
<type>	function of the eighth upstream receiver; one of: • **standard** — use the receiver as a standard upstream • **spec-analysis-only** —use the receiver only for ingress avoidance spectrum analysis • **redundant-upstream-only** — use the receiver for redundant upstream channel operation • **spec-analysis-and-redundant** —use the receiver for both ingress avoidance spectrum analysis and redundant upstream channel operation (if the receiver switches, either automatically or under operator control, this disables ingress avoidance spectrum analysis until you disable the redundant switch) • **spec-analysis-debug-only**—use the receiver for spectrum analysis under manual control (see “Using the spectrum analyzer for troubleshooting” on page 10-17)

2 Type **exit** to return to the Console> prompt.

—continued—

Procedure 3-3 (continued)
Configuring the eighth upstream receiver using the CLI

Manually switching the upstream to the redundant receiver

1 Switch a failed upstream to the receiver in slot 8 using the following command:

[] Console> **upstream-failover <channel>** ↵

where

<channel>	is the number of the failed upstream (**1** to **7**), or **0** to cancel a previous switch

Note 1: Any modems attached to the eighth channel are disconnected. If you plan to use this feature, do not attach modems to the eighth channel.

Note 2: If you are using the eighth upstream receiver for spectrum analysis, switching the receiver to another channel disables spectrum analysis. You must cancel the switch using **upstream-failover 0** to restore the spectrum analyzer function.

—end—

Procedure 3-4
Configuring the eighth upstream receiver using SNMP

The Cornerstone CMTS provides special functionality for an upstream receiver module installed in slot 8. You can configure the receiver to operate in any of the modes listed in Table 3-3 on page 3-13.

> ***Note:*** Under certain circumstances, the spectrum analyzer function may stop working. If this occurs, disable the spectrum analysis feature, then re-enable the feature.

Use the lcCmtsMultiUsageUs MIB objects to set the function for this upstream receiver.

Action

Perform the following tasks in any order when needed.

Task	Page
Configuring the receiver function	3-17
Manually switching an upstream to the redundant receiver	3-18
Viewing the current settings	3-18

Configuring the receiver function

1 Change the lcCmtsMultiUsageUsAdmin object to configure the upstream receiver function as follows:

If you want to use the receiver as…	**Then** set the object to…
a normal upstream receiver	Standard (0)
a spectrum analyzer	IngressAvoidanceSpectrumAnalysis (1)
a redundant receiver	RedundantUpstream (2)
both a spectrum analyzer and a redundant receiver	IngressAvoidanceAndRedundant (3)
a spectrum analyzer without ingress avoidance	SpectrumAnalysisDebugOnly (4)

—continued—

Procedure 3-4 (continued)
Configuring the eighth upstream receiver using SNMP

Manually switching an upstream to the redundant receiver

1 To switch a failed upstream to the receiver in slot 8, set the lcCmtsRedundancyFailover object to the number of the failed upstream (**1** through **7**).

2 To cancel a switch, set the lcCmtsRedundancyFailover object to **none** (or **0**).

Viewing the current settings

1 To view the current settings for the receiver, read the lcCmtsMultiUsageUsOperStatus object. The status is as follows:

If the value is…	**Then** the receiver is…
standard	a standard upstream receiver
ingressAvoidanceSpectrumAnalysis	a spectrum analyzer
redundantUpstream	a redundant module (see step 2)
spectrumAnalysisDebug	a spectrum analyzer (not used for ingress avoidance)
frontPanelRedirect	redirected to the front panel test port

2 To view the current protection status of the receiver, when configured as a redundant receiver, read the lcCmtsRedundancyFailover object.

If the value is…	**Then**
none (0)	the receiver is not currently providing redundant service
1 through **7**	the receiver has been switched, manually or automatically, to a failed upstream

—end—

Controlling and managing user access

The Cornerstone CMTS is an important part of your data network and must be secured against both malicious attacks from outside and mistakes by untrained personnel. This chapter shows you how to configure access for users and network management systems.

In this chapter

This chapter contains the procedures listed in Table 4-1.

Table 4-1
Procedures in this chapter

Procedure	Title	Page
4-1	Creating and modifying user accounts using the CLI	4-7
4-2	Configuring access to SNMPv1 objects in NmAccess Mode	4-11
4-3	Modifying network manager access using SNMPv1	4-14
4-4	Configuring SNMPv3 access and privileges in Coexistence Mode	4-17
4-5	Using SSH to provide secure remote access	4-23

Access methods

You can access the CMTS using the following methods:

- the console port for CLI access
- telnet or SSH (the CMTS supports up to five simultaneous telnet or SSH sessions plus a single session from the console port) through the Ethernet and cable interfaces
- SNMP (for network management systems and provisioning servers); the CMTS 1500 supports coexistence of multiple SNMP communities: SNMP v1/v2/v3.

Terms and concepts

This chapter uses the following terms and concepts:

Community string
A character string, similar to a password, that allows access to one or more network objects by a network management system.

Object identifier (OID)
The internal representation of a MIB object. This is a string of decimal numbers separated by periods, such as **1.3.6.1.2.1.1.1.4.1**.

Authentication protocol
A mechanism in SNMPv3, used to guarantee authorization to individual users.

Privacy protocol
A mechanism in SNMPv3, used to encrypt data traffic between a network management system and the managed device.\

NmAccess Mode
The default mode of the CMTS that supports SNMPv1 and SNMPv2 protocols via the docsDevNmAccessTable.

Co-existence
Supports the simultaneous use of SNMPv1 and SNMPv2 applications to the more secure SNMPv3 environment.

Default access (direct login and SNMPv1)

This section briefly describes the available user accounts and security features.

Default user accounts and passwords for logging into the Console,Telnet, or SSH session

The default user account name is **root**. This account has read/write privileges.

The default user account password is blank (press **Return**). You should change this password immediately to prevent unlimited access by unauthorized people.

Default community strings for NmAccess Mode

The initial entry in the docsDevNmAccessTable has a null community string.

For each newly-created entry in the docsDevNmAccessTable, the default community string is “public.”

Network management access

The docsDevNmAccessTable MIB object provides a basic level of security and control of network management access to the CMTS. The basic concept is that one (or more) super-users (“Account Manager”) have top-level control of the system, and can create or delete other user accounts. These user accounts are given appropriate user privileges, and can have their own separate community strings and passwords.

You can manage network management access through the CLI, the SNMP manager, or a provisioning program such as the DOCSIS LCn or CPS 2000. This chapter covers the CLI and SNMP Manager methods.

SNMP Support

The ARRIS CMTS SNMP agent supports SNMPv1, SNMPv2, and SNMPv3 protocols.

SNMPv3 support is a requirement for DOCSIS 1.1. It provides more flexible and secure authentication than SNMPv1 and SNMPv2. It also defines the security for management access.

SNMPv3 allows a network administrator to perform the following:

- Set up user accounts.
- Provide strong authentication of SNMP traffic (optional).
- Configure encryption to ensure SNMP private transactions.

In addition, portions of the MIB tree can be restricted for a user, by using the SNMPv3 protocol to provide “view-based” access.

SNMPv3 uses the User-based Security Model (USM) to control access to MIBs and network devices. The USM consists of four major parts:

- Users—individual login accounts. In some instances, the user ID is referred to as a “security name.”
- Groups—defines access rights for one or more users.
- Access Table—specifies the views used for read access, write access, and access to SNMPv3 notifications.
- Views—specifies which MIB objects are (or are not) available to a user.

You can provision access to cable data network objects by both SNMPv1 and SNMPv3 clients.

Co-existence

Co-existence supports the migration of SNMPv1 and SNMPv2 applications to the more secure SNMPv3 environment. It also supports aspects of SNMPv3 security for SNMPv1 and SNMPv2 access.

To be able to switch to co-existence mode, a user must have **public_v1** or **public_v2** as the default community name in the SNMP community list. Refer to Procedure 4-6 for additional information.

The generation of traps and notifications, with the support of co-existence, is subject to rigorous security verification. The network administrator chooses whether SNMPv1/v2 traps or notifications inform with or without SNMPv3 security fields generated by the CMTS.

The CMTS 1500 is fully DOCSIS 1.1 compliant. It supports the co-existence mode defined in the DOCSIS 1.1 OSSI specification. When the SNMPv3 environment is not appropriate, the CMTS supports standard SNMPv1 and SNMPv2 protocols via the docsDevNmAccesstable. This access is referred to as “NMAccess mode”.

SNMPv3 data flow

The following steps describe how a device supporting SNMPv3 handles incoming requests. Information *in italics* describes how the device processes an error.

1 The device receives an SNMPv3 packet containing a user’s name and security parameters.

2 If privacy is enabled, the device decrypts the data.

If decryption failed, return an error.

3 The device authenticates the user name.

If the user name does not appear in the list of valid users, return an error.

4 The device finds the user name in the Group table.

If the user name is not in any group, return an error.

5 The device looks up the group’s views in the Access table.

If the group does not appear in the Access table, return an error.

6 The device checks the object against the view, and takes the appropriate action (for example, set or retrieve the value) if the view indicates that this object is accessible to the user.

If the object is not accessible, return an error.

7 The device returns a valid response.

Special considerations

CAUTION
All operators can be locked out
The following actions can prevent all further access to the CMTS:

- deleting all account manager accounts
- deleting the last privilege entry
- changing the NmAccess table to prevent access by all operators

If the CMTS is not allowing access for these reasons, contact your next level of support.

Use of subnet masks in account management

Certain account management commands take an IP address and subnet as parameters. These commands use that information differently from the standard IP routing commands.

The subnet mask defines which part of an IP address must match the specified host address for the CMTS to allow access. For example, if the host IP address is **10.42.69.3** and the subnet mask is **255.255.255.0**, then any host with an IP address in the range **10.42.69.0** to **10.42.69.255** would be allowed access.

Procedure 4-1
Creating and modifying user accounts using the CLI

The CMTS supports up to 10 user accounts. The following procedure provides details for viewing, creating, modifying, and deleting accounts.

Requirements

You must be logged in as **root** (or any another account that has account management privileges). The tasks in this procedure assume that you are starting from the top-level (Console> or Remote>) prompt.

Action

Perform the tasks in this procedure in the specified order.

Task	**Page**
Creating and modifying accounts	4-8
Creating or modifying a privileges profile	4-9
Viewing accounts currently logged in	4-10
Viewing all user accounts	4-10

—continued—

Procedure 4-1 (continued)
Creating and modifying user accounts using the CLI

Creating and modifying accounts

1 Enter the following commands to move to the “accounts” level:

```
Console> manage ↵
[] box# accounts ↵
[] accounts# show user-list ↵
```

The CMTS displays the parameters and values for the configured accounts.

2 Enter the following commands to select an account:

```
[] accounts# user-modify/<#> ↵
[] user-modify/# info ↵
```

where	
<#>	is the account number to modify: **1** to **10**

The CMTS displays the user name parameters and values.

3 Enter the following commands to modify the selected account. The examples assume that you are modifying account 3.

```
[] user-modify/3# user-name “<userid>” ↵
[] user-modify/3# privileges-index <index> ↵
[] user-modify/3# status active ↵
```

where	is the…
<userid>	login name for this account
<index>	index number of the privileges profile you want for this account (see “Creating or modifying a privileges profile” on page 4-9)

4 Type **exit** to return to the top-level Console> prompt.

—continued—

Procedure 4-1 (continued)
Creating and modifying user accounts using the CLI

Creating or modifying a privileges profile

1 Enter the following commands to change a privileges profile:

[] accounts# **privileges-modify/<#>** ↵

[] privileges-modify/# **info** ↵

where	
<#>	is the privileges profile to modify: **1** to **10**.

The CMTS displays the values assigned to the selected privileges profile.

2 Enter the following commands to modify the selected profile. The examples assume that you are modifying profile 3.

[] privileges-modify/3# **level-name <level>** ↵

[] privileges-modify/3# **security-name <community>** ↵

[] privileges-modify/3# **status active** ↵

where	is the…
<level>	character string describing this account
<community>	SNMP community string assigned to this profile. ***Note:*** The community string specified here must match one of the configured community strings.

3 Type **exit** to return to the top-level Console> prompt.

—continued—

Procedure 4-1 (continued)
Creating and modifying user accounts using the CLI

Viewing accounts currently logged in

1 Enter the following command at the Console> prompt:

who ↵

The CMTS displays the following information for each account:

- Session—Console or Remote[1-5]
- User name—User name used at login
- Location—IP address (**127.0.0.1** indicates the Console port)
- RemPort—Remote port (**None** is default for Console)
- Interface—**UART** (for Console), **CATV**, or **Ethernet**
- Start of session—The date and time the current session started.

Note: The **who** command displays only user accounts that are logged in.

Viewing all user accounts

1 Enter the following commands to display all user accounts:

Console> **manage** ↵

[] box# **accounts** ↵

[] accounts# **show user-list** ↵

The CMTS displays a table showing all configured accounts.

The table has four columns, as described below. All CMTS units come with one default account (root), assigned to index 1.

- Index—Index number (1 to 10) assigned by CMTS to the User Account. A maximum of 10 accounts can be set up.
- User Name—user name for CMTS login. Avoid using a blank user name, to avoid confusion.
- Privileges Index—This value corresponds to the index value used in the Account Privileges table (displayed by the **show privileges-list** command).
- Status—Displays current status; the account must be **active** to be used.

2 Type **exit** to return to the top-level Console> prompt.

—end—

Procedure 4-2
Configuring access to SNMPv1 objects in NmAccess Mode

Use this procedure to grant or restrict access to SNMP objects by specific hosts, using SNMPv1 protocol in NmAccess Mode.

Requirements

You must be logged into the CMTS CLI as **root** (or any another account that has account management privileges).

This procedure assumes that you are at the top-level Console> (or Remote>) prompt. If not, type **exit** to return to the top-level prompt.

CAUTION
Possible loss of access
Making changes to administrative accounts can result in losing all root-level access to the CMTS. Be careful when changing these accounts.

Default community strings

For each newly-created entry in the docsDevNmAccessTable, the default community string is "public."

Action

Follow these steps to configure SNMP access.

1 Enter the following commands to access the SNMP level:

Console> **manage** ↵

[] box# **snmp** ↵

[] snmp# **show** ↵

The CMTS displays the next level show and info.

2 Enter the following commands to view the Access List table:

[] snmp# **show access-list** ↵

The CMTS displays the current access list.

—continued—

Procedure 4-2 (continued)
Configuring access to SNMPv1 objects in NmAccess Mode

3 Enter the following commands to view the access-specific parameters:

```
[] snmp# nmaccess ↵
[] snmp# access-specific/<index> ↵
[] access-specific/3# info ↵
```

where	
<index>	is the index of an access profile: **1** to **9**, or **2147483647** to indicate the default system administrator account

The CMTS displays the current values for the selected profile.

4 Enter the following commands to modify the selected access profile:

```
[] access-specific/3# ip-addr <hostip> ↵
[] access-specific/3# ip-mask <hostsub> ↵
[] access-specific/3# community <string> ↵
[] access-specific/3# control <level> ↵
[] access-specific/3# interfaces <devices> ↵
[] access-specific/3# extensions <privs> ↵
[] access-specific/3# status <status> ↵
```

where	is the…
<hostip>	IP address of the host system
<hostsub>	see “Use of subnet masks in account management” on page 4-6
<string>	community string this host uses
<level>	level of control allowed this host: • **none**—all access is denied • **read-only**—the host cannot change data • **read-write**—this host can change data • **ro-with-traps**—read-only; the host receives SNMP traps • **rw-with-traps**—read-write; the host receives SNMP traps • **traps-only**—no access; the host receives traps

—continued—

Procedure 4-2 (continued)
Configuring access to SNMPv1 objects in NmAccess Mode

where	is the…
<devices>	list of interfaces that can be controlled: • **ethernet** • **cable** • **ethernet+cable**
<privs>	extra privileges for this access profile; one of: • **none** (default) • **reset-allowed** • **account-manager**
<status>	current status of this profile: • **active**—the profile is active • **not-in-service**—the profile exists, but is not applied to the host • **delete**—deletes this profile

5 Type **info** to verify the settings for the profile.

6 Type **exit** to return to the Console> prompt.

—end—

Procedure 4-3
Modifying network manager access using SNMPv1

User account management via SNMP is performed through two MIB object tables:

- DocsDevNmAccessTable—this table provides a basic level of security for access to the devices by network managers.

  Note: If the DocsDevNMAccessTable is empty (no entries) access to SNMP objects is unrestricted.

- DocsDevNmAccessExtensionTable—this table controls access to SNMP objects by network management stations. If the table is empty, access to SNMP objects is unrestricted.

The accounts can be managed through use of an SNMP manager, or by CLI Get and Set commands to the MIB objects and instances.

Requirements

You must have one of the following:

- access to an SNMP manager with read/write access and account management privileges
- an account on the CMTS with read/write and account management privileges

See “Special considerations” on page 4-6 for considerations, especially when deleting accounts or modifying system-wide account parameters.

Default community strings

The initial entry in the docsDevNmAccessTable has a null community string.

For each newly-created entry in the docsDevNmAccessTable, the default community string is “public.”

—continued—

Procedure 4-3 (continued)
Modifying network manager access using SNMPv1

Action

1 Create or modify an entry in the DocsDevNmAccessTable and set the objects as follows:

Object	**Description**
docsDevNmAccessEntry	Defines the entries in the table.
docsDevNmAccessIndex	The index value for the table, which orders the application of access entries.
docsDevNmAccessIp	The IP address (or subnet) of the network management station (NMS). If traps are enabled for this entry, then the value must be the address of a specific device.
docsDevNmAccessIpMask	See "Use of subnet masks in account management" on page 4-6. If traps are enabled for this entry, then the value must be **255.255.255.255**.
docsDevNmAccessCommunity	The community display string to be matched for access by this entry. If set to the null string (""), then any community string will match. The default is "public" for new accounts.
docsDevNmAccessControl	Specifies the access type allowed to this NMS. Account Managers should be very careful when setting this MIB object. The default value is **2** (read).
docsDevNmAccessInterfaces	Specifies the set of interfaces from which requests from this NMS will be accepted. ***Note:*** This object only applies to the link-layer interfaces (Ethernet and CATV MAC).
docsDevNmAccessStatus	Controls and reflects the status of rows in this table.

For detailed descriptions of each object, see the *MIB Reference* or the descriptions available through the SNMP manager.

Note: Access is also constrained by the community strings and the BPI+ objects.

—continued—

Procedure 4-3 (continued)
Modifying network manager access using SNMPv1

2 Open the entry in the lcNmAccessExtensionTable that corresponds to the entry you are modifying in the DocsDevNmAccessTable and set the following object:

Object	**Description**
lcNmAccessAdditionalPrivileges	Defines additional privileges allowed a user. This object represents each additional privilege as a bit; you can set or clear each bit individually. • **resetAllowed** (0x80)—allows an operator with a read-only account to reset the CMTS from either a telnet or console session. It does not allow the user to use the **reset factory** command (which requires read/write access), nor does it allow remote SNMP write access to either the docsDevResetNow or the lcRestartFromFactoryDefault MIBs. A read/write entry gets reset ability automatically, so setting **resetAllowed** for read/write entries is not necessary. • **accountManager** (0x40)—allows a read/write account to create, modify, and delete accounts. Read/only accounts ignore the **accountManager** bit.

—end—

Procedure 4-4
Configuring SNMPv3 access and privileges in Coexistence Mode

Use this procedure to configure the SNMPv3 access and privileges in Coexistence mode.

Action

Perform the tasks in this procedure in any order.

Task	Page
Creating a new SNMPv3 user	4-18
Creating a new SNMPv3 group	4-20
Creating an SNMPv3 view	4-21
Creating entries in the access table	4-22

—continued—

Creating a new SNMPv3 user

1 At the CLI Console> prompt, enter the following commands:

Console> **manage** ↵

[] box# **snmp** ↵

[] snmp# **coex** ↵

[] coex# **ver3** ↵

[] ver3# **show v3user-list** ↵

The CMTS displays a list of configured users.

2 Set up a new user by entering the following commands:

[] ver3# **v3user-modify/<userid>** ↵

[] v3user-modify/userid# **clone <template>** ↵

where	is the…
<userid>	name of the user you want to configure.
<template>	name of the user ID whose configuration you want to copy to the new user ID. The CMTS includes predefined user IDs for use as templates, including: • **templateMD5** (uses MD5 authentication) • **templateSHA** (uses SHA authentication and DES privacy) • **initial** (limited SNMPv3 security, no authorization, no privacy) • **dhKickstart** (basic features with diffiehelman-based security mechanisms)

—continued—

Procedure 4-4 (continued)
Configuring SNMPv3 access and privileges in Coexistence Mode

3 (optional) Change the authorization and privacy protocols for the new user by entering the following commands:

```
[] v3user-modify/userid# authorization-protocol <authproto> ↵
[] v3user-modify/userid# privacy-protocol <privproto> ↵
```

where	is the…
<authproto>	authorization protocol you want to use; one of: • **usmNoAuthProtocol** (can be set only if privacy is already set to **usmNoPrivProtocol**) ***Note:*** The only way to specify an authorization protocol for a user is to clone a user with the protocol that you want to use.
<privproto>	privacy protocol you want to use; one of: • **usmNoPrivProtocol** ***Note:*** The only way to specify an privacy protocol for a user is to clone a user with the protocol that you want to use.

4 Change the authorization password for the new user by entering the following commands:

```
[] v3user-modify/userid# @v3p -a userid ↵
old password:
new password:
re-enter new password:
[] v3user-modify/userid#
```

5 Change the privacy password for the new user by entering the following commands:

```
[] v3user-modify/userid# @v3p -p userid ↵
old password:
new password:
re-enter new password:
[] v3user-modify/userid#
```

Note: The default password for authorization and privacy is **maplesyrup.** You must change the default password for any account using authorization or privacy before it can be enabled.

6 Activate the new user account by entering the following command:

```
[] v3user-modify/userid# status active ↵
```

7 Type **exit** to return to the Console> prompt.

—continued—

Procedure 4-4 (continued)
Configuring SNMPv3 access and privileges in Coexistence Mode

Creating a new SNMPv3 group

1 At the CLI Console> prompt, enter the following commands:

Console> **manage** ↵

[] box# **snmp** ↵

[] snmp# **coex** ↵

[] coex# **ver3** ↵

[] ver3# **show group-list** ↵

The CMTS displays a list of configured groups.

2 Set up a new group by entering the following command:

[] ver3# **group-modify/<snmp model>/<userid>** ↵

where	is the…
<snmp model>	SNMP version: **v1**, **v2**, or **USM** (v3)
<userid>	name of the user you want to assign to the new group.

3 Create a new group and assign it to the selected user by entering the following command:

[] group-modify/usm/userid# **group "<groupname>"** ↵

where	is the…
<groupname>	name of the group you are creating.

4 Type **exit** to return to the Console> prompt.

—continued—

Procedure 4-4 (continued)
Configuring SNMPv3 access and privileges in Coexistence Mode

Creating an SNMPv3 view

1 At the CLI Console> prompt, enter the following commands:

Console> **manage** ↵

[] box# **snmp** ↵

[] snmp# **coex** ↵

[] coex# **ver3** ↵

[] ver3# **show view-list** ↵

The CMTS displays a list of configured views.

2 Create a new view by entering the following command:

[] group-modify/userid# **view-modify/<viewname>/<oid>** ↵

where	is the…
<viewname>	name of the view you are creating.
<oid>	MIB object to use as the root of the view's subtree; one of: • an object identifier such as **1.3.6.1.2.1.1** • a MIB name such as **docsDevSoftware**

3 Set the mask for the new view by entering the following command:

[] view-modify/viewname/oid# **mask <octets>** ↵

where	is the…
<octets>	series of octets, in decimal, that specify the mask to use with this view. The mask allows partial matches between the view and the user-specified object. For example, if the view's object identifier (OID) is **1.3.6.1.2.1.1.4.0** and the mask is **1**, then the view matches **1.3.6.1.2.1.1.1.4.1**, but not **1.3.6.1.2.1.1.5.0**. A value of **1** is equivalent to no mask. A value of **0** allows anything to match.

4 Repeat steps 2 and 3 for each subtree you want to add to the view.

5 Set the type for the new view by entering the following command:

[] view-modify/viewname/oid# **type included** ↵

6 Type **exit** to return to the Console> prompt.

—continued—

Procedure 4-4 (continued)
Configuring SNMPv3 access and privileges in Coexistence Mode

Creating entries in the access table

1 At the CLI Console> prompt, enter the following commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# coex ↵
[] coex# ver3 ↵
[] ver3# show v3access-list ↵
```

The CMTS displays the access table for each group and entry.

2 Create a new access table entry by entering the following command:

```
[] ver3# v3access-modify/<group>/<snmp model>/<sec> ↵
```

where	is the…
<group>	name of the group using this access entry
<snmp model>	SNMP version: **v1**, **v2**, or **USM** (v3)
<sec>	security level; one of: • **noAuthNoPriv**—no authentication or privacy • **authNoPriv**—with authentication but without privacy • **authPriv**—with authentication and privacy

3 Assign views to the access entry by entering the following commands:

```
[] v3access-modify/group/usm/sec# read-view-name "<view>" ↵
[] v3access-modify/group/usm/sec# write-view-name "<view>" ↵
[] v3access-modify/group/usm/sec# notify-view-name "<view>" ↵
```

where	is the…
<view>	name of the view to assign to each access type. Note that you can assign different read, write, and notify views to an access entry.

4 Repeat steps 2 and 3 for each access entry you want to add to the group.

5 Type **exit** to return to the Console> prompt.

—end—

Procedure 4-5
Using SSH to provide secure remote access

SSH (Secure Shell) provides operators with encrypted access to the CMTS. This is important for remote access, since potential attackers often monitor such links looking for passwords or other privileged information.

> *Note:* SSH does not provide extra account access security; anyone with an SSH client and a valid CMTS user account and password can access the CMTS through SSH. SSH only provides a encrypted link that prevents password “sniffing.”

SSH version supported

The CMTS 1500 currently supports only SSH v1 access. SSH v2 is not compatible with SSH v1.

Terms and concepts

The tasks in this procedure use the following terms and concepts.

Key

A block of bits, used to encrypt or decrypt data on the SSH link. Keys can be from 512 to 2048 bits long, in increments of 128 bits. Longer keys are harder to decrypt by someone attempting to compromise your security, but can seriously impact CMTS performance.

Client

A system that a remote operator uses to connect to the CMTS using SSH.

Host

The CMTS.

Host key

Actually a pair of keys, one public (sent to clients) and one private (not distributed) that the CMTS uses to decrypt incoming traffic.

Server key or session key

A key that the CMTS creates for each SSH session to encrypt outgoing traffic. This key is regenerated periodically for added security.

Key sizes

The host key and server key must have different sizes; the difference must be at least 128 bits.

The recommended key sizes are as follows:

- host key: 768 bits
- server key: 512 bits

These key sizes provide commercial-grade encryption. Larger key sizes require a great deal of time and CMTS processing resources (a 2048-bit key can take 15 minutes when the CMTS is doing nothing else).

> ***Note:*** The values for the host key and server key cannot be entered via the CLI separately. Both parameters must be entered and keys regenerated at the same time. If one parameter is entered after the other, the CLI forgets what was entered first.

Action

Perform the tasks in this procedure in any order.

Task	Page
Initializing SSH support from the CLI	4-25
Generating and regenerating host keys	4-26
Configuring and controlling SSH access through SNMP	4-27

—continued—

Procedure 4-5 (continued)
Using SSH to provide secure remote access

Initializing SSH support from the CLI

1 Enter the following commands, starting at the CLI Console> prompt, to access the administration level:

Console> **manage** ↵

[] box# **admin** ↵

The CMTS displays the admin# *prompt.*

2 Enter the following commands to enable and configure SSH support:

[] admin# **ssh-host-key-bits <hostsize>** ↵

[] admin# **ssh-server-key-bits <srvsize>** ↵

[] admin# **ssh-key-regen-mins <time>** ↵

[] admin# **ssh-control enabled** ↵

where	is the…
<hostsize>	size, in bits, you want to use for the host key. Range: **512** to **2048** Default: **768**
<srvsize>	size, in bits, you want to use for the server key. Range: **512** to **2048** Default: **512**
<time>	interval, in minutes, that the CMTS waits before regenerating the session key Range: **5** to **10080** (1 week) Default: **5**

3 Proceed to “Generating and regenerating host keys” on page 4-26.

4 Type **exit** to return to the Console> prompt.

—continued—

Procedure 4-5 (continued)
Using SSH to provide secure remote access

Generating and regenerating host keys

1 Proceed as follows:

If you want to…	**Then** go to…
show the current keys	step 2
regenerate the host keys with a new key size	step 3
change the session key size	step 4

2 Enter the following command from the CLI Console> prompt to view the host key sizes:

Console> **ssh-keygen** ↵

The CMTS shows the current key size.

3 Enter the following command to generate new host keys with a new key size:

Console> **ssh-keygen -hb <size>** ↵

where	is the…
<hostsize>	size, in bits, you want to use for the host key. Range: **512** to **2048**

The CMTS generates new host keys.

4 Enter the following command to generate new session keys with a new key size:

Console> **ssh-keygen -sb <size>** ↵

where	is the…
<hostsize>	size, in bits, you want to use for the session key. Range: **512** to **2048**

The CMTS changes the session key size but does not generate new host keys.

Note: It is recommended you enter the commands for the new host key and session key on the same line otherwise entering each command separately deletes the previous command. For example:

Console> **ssh-keygen -hb <size> -sb <size>** ↵

5 Enter the following command to regenerate the host key:

Console> **ssh-keygen -regen** ↵

The CMTS regenerates the host key using the most recent size parameters.

6 Type **exit** to return to the Console> prompt.

—continued—

Procedure 4-5 (continued)
Using SSH to provide secure remote access

Configuring and controlling SSH access through SNMP

1 Open the lccmtsSSH1KeyParams group and set the objects as follows:

Object	Description
lccmtsSSH1KeyParamsControl	Enables or disables SSH support; one of: • **enabled** • **disabled**
lccmtsSSH1KeyParamsHostBits	The size, in bits, of the host public key: **512** to **2048**.
lccmtsSSH1KeyParamsServerBits	The size, in bits, of the server key: **512** to **2048**.
lccmtsSSH1KeyParamsServerKeyRegenMins	The time, in minutes, that the CMTS waits before generating a new session key: **5** to **10080**.

—end—

Using filters to manage network traffic

This chapter describes how to configure filters, traps, and notifications using the CLI or the SNMP manager.

Filters are used to pass or block data packets at the CMTS or cable modem (RF or Ethernet). Filters are first configured, and then activated or deactivated as desired.

Filtering in the CMTS

Note: Order of filters is VERY important.

The CMTS supports multiple filters, allowing users to precisely filter packets to meet network and traffic requirements.

The CMTS has two main filter types:

- Logical Link Control (LLC) Layer filters
- IP filters

The CMTS also has a number of other packet selection and control features which resemble filters in their implementation and operation, such as Proxy ARP filtering.

Logical Link Control (LLC) layer filters

The LLC, or Logical Link Control layer, operates in the upper portion of the data link layer in the CMTS. This layer manages the data-link communication, and link-addressing functions. The Service Access Points (SAPs), which define the LLC protocol in use, are specified at this level. The data-link communications can be connectionless or connection-oriented.

LLC filters are used only for inbound packets. You can configure LLC filters to block or pass entire services or protocols, based on the SAP or EtherType. The LLC filters restrict forwarding by network-layer protocols (such as IP, IPX, NetBIOS, and Appletalk).

Examples of protocols and services are:

- 0x06 (IP) SAP
- 0xE0 (Novell) SAP
- 0x0800 (IP) EtherType
- 0x0806 ARP
- 0x8137 (Novell) EtherType

LLC filters have a default action setting: either block or pass ALL unmatched packets. These filters are very useful in eliminating or blocking unwanted protocol traffic on the network.

IP filters

The CMTS supports IP filters with many tuneable parameters per IP filter.

IP filters (like LLC filters) have a default action (pass or block) setting, for packets that do not match any IP filter. In addition, you can configure individual IP filters to pass or block packets that match the filter.

The most common IP Port filter parameters are:

IP address filters

The primary IP address filters are:

- Source address
- Source mask
- Destination address
- Destination mask

The subnet mask defines which part of an IP address must match the specified host address for the filter to match. For example, if the filter IP address is **10.42.69.3** and the subnet mask is **255.255.255.0**, then any host with an IP address in the range **10.42.69.0** to **10.42.69.255** would match.

IP port and IP protocol filters

IP port and protocol filters are used to block specific ports or services in the IP protocol suite. These filters operate at the Layer 3 level. Examples of IP Port filters are:

- Port 69 (TFTP) connectionless UDP
- Port 80 (WWW) connection-oriented TCP
- Port 21 (FTP)

Other IP filters

The CMTS has other filters for controlling items such as:

- Policy
- Broadcast

Differences between LLC filters and IP filters

LLC filters block the entire protocol service. You can also use LLC filters to block TCP/IP traffic.

IP filters block specific IP addresses, address ranges, ports (services) and protocols within the IP protocol suite.

Typical filter applications

Some typical filter applications might be:

- Use LLC filters to block all Novell traffic from the cable modem network.
- Use IP Port filtering to prevent subscribers from running a web server or FTP server on the cable modem network.

Multicast filtering

Multicast filters (IGMPv2 and ICMP Type 9) have distinct characteristics.

IGMPv2 (multicast) filters

The CMTS implements IGMP as a proxy operation, and has two sides: router and host. The CMTS generates replies to queries received from the Ethernet interface and generates queries on the CATV interface. This allows the CMTS to keep multicast traffic to a minimum on the cable network.

Any IP-Multicast packet received on the CMTS CATV interface will be filtered if it does not appear in the IGMP address table (maintained by the CMTS).

Multicast addresses are forwarded to CATV interface if the IGMP proxy is disabled, or if the IGMP address is found in the IGMP cache table, If IGMP proxy is enabled and the multicast address is not found in the cache table, the multicast packets are discarded. If there is a static multicast entry in the CMTS, the CMTS will forward multicast traffic for that address regardless of forwarding state.

ICMP Type 9

ICMP Type 9 is the router advertisement message necessary for routers implementing multicasting. The CMTS can be configured to listen to these messages or ignore them. Additional information can be found at http://www.networksorcery.com/eng/protocol/icmp/msga.htm.

Proxy ARP

The purpose of Proxy ARP is to help reduce the amount of total traffic that goes across the cable plant. It was not designed for security purposes such as preventing subscribers from learning upstream and routing paths.

Proxy ARP works as follows: If proxy ARP is enabled, any ARP responses that the CMTS learns on the cable interface side is stored for later usage. If a node on the Ethernet side of the CMTS tries to ARP for a node that is on the cable side of the CMTS and the CMTS has a learned entry for that node, the CMTS responds to that ARP request on behalf of the node that is behind the cable side of the CMTS. This helps reduce the amount of unnecessary traffic going across the cable plant and preserves the bandwidth.

The CMTS must have an entry in its own ARP tables about the node on the cable side of the CMTS. If it does not, then the ARP process operates as normal with the CMTS merely passing along the traffic without interfering with it until it learns of the ARP entry on the cable side.

When the CMTS responds to the ARP request coming from the Ethernet side, the CMTS makes the ARP response look like it came from a node on the cable side. It does not rewrite the MAC address of the ARP response as its own.

When the CMTS is configured for Proxy ARP (see “Configuring Proxy ARP filters” on page 5-19), the CMTS does not forward ARP requests destined to cable modems or CPE devices to the CMTS cable interface.

Note: The CMTS does not support Static ARP entries.

Application of filters (filtering order)

In general, filters are applied in the following order:

1 Inbound LLC layer filters

2 IP Port inbound filters

3 IP Port outbound filters

Other filters can be applied in any desired order.

Cable modem filtering

The CMTS can control filtering in the cable modem at the RF port or at the Ethernet port. The filtering capabilities of cable modems varies, depending upon the manufacturer and software. Cable modems do not perform IP processing (except filtering), so the IP inbound and IP outbound filters generally can be combined.

Special Internet addresses

A number of Internet addresses have been reserved for special uses; you may wish to filter them to avoid propagation. The most common special addresses are:

- 0.0.0.0/8
- 127.0.0.0/8
- 192.0.2.0/24
- 10.0.0.0/8 (Refer to RFC 1918)
- 172.16.0.0/12 (Refer to RFC 1918)
- 192.168.0.0/16 (Refer to RFC 1918)
- 169.254.0.0/16

For more information regarding these addresses, refer to the IETF draft: draft-manning-dsva-ob.txt.

Procedure 5-1
Configuring filters using the CLI

The CMTS filtering can be controlled through the following CLI **manage** sub-commands:

- **ethernet-level**
- **cable-level**
- **forwarder**
- **ip-level**

Note: You can reset (destroy) all filters by setting the lcResetFilters MIB object to **true**. You can use the CLI **set** command to do this as follows:

```
[] Console> set lcResetFilters.0 true ↵
```

Requirements

You must be logged into the CMTS CLI with administrative privileges.

Action

Perform the following tasks in this procedure in any order.

Task	**Page**
Configuring Link Layer Control (LLC) filters	5-8
Configuring IP filters	5-10
Configuring IGMPv2 filters	5-13
Configuring ICMP Type 9 Router Advertisement Messages filters	5-14
Displaying and configuring port filters	5-14
Configuring Spanning Tree filters	5-18
Configuring Proxy ARP filters	5-19

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

Configuring Link Layer Control (LLC) filters

1 Set the action to take on filtered packets using the following commands:

Console> **manage** ↵

[] box# **forwarder** ↵

[] forwarder# **llc-filter-default <action>** ↵

where

<action> is the action to take on packets that match any LLC filter:

- accept (forward filtered packets)
- discard (discard filtered packets)

2 Display the current list of LLC filters using the following commands:

Console> **manage** ↵

[] box# **forwarder** ↵

[] forwarder# **link-filter-list** ↵

[] link-filter-list# **show** ↵

The CLI displays the currently configured LLC filters.

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

3 Create or modify an LLC filter using the following commands:

```
[] link-filter-list# back ↵
[] spanning-tree# link-filter-modify/<index> ↵
[] link-filter-modify/3# interface <port> ↵
[] link-filter-modify/3# type-of-protocol <type> ↵
[] link-filter-modify/3# layer-3-protocol <llcproto> ↵
[] link-filter-modify/3# status <filterstat> ↵
```

where	is the…
<index>	number of the filter you want to configure. You can create up to 10 LLC filters.
<port>	port to monitor; one of **ethernet**, **cable**, or **ether-and-cable**
<type>	protocol format; one of **ethertype** (a two-byte Ethernet type or SNAP-encapsulated frames) or **dsap** (a one-byte 802.2 type).
<llcproto>	hexadecimal number that indicates the protocol to filter. If <type> is **ethertype**, typical values are: • **0x8000** — IP • **0x8006** — IP ARP • **0x809B** — Appletalk • **0x80F3** — Appletalk ARP • **0x8137** — IPX Ethernet Frame If <type> is **dsap**, typical values are: • **E0** — IPX 802.2/802.3 Frame • **F0** — Netbios Frame
<filterstat>	filter status; one of the following: • **create-and-go** to create and activate a new filter • **create-and-wait** to create a new filter without activating it • **active** to activate an inactive filter • **delete** to remove a configured filter • **not-in-service** to deactivate a filter but not delete it

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

4 Repeat step 3 for each LLC filter you want to configure.

5 Type **exit** to return to the Console> prompt.

Configuring IP filters

1 Display the current list of IP filters using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# ip-filter-list ↵
[] ip-filter-list# show ↵
```

The CLI displays the currently configured IP filters.

2 Create or modify an IP filter using the following commands:

```
[] ip-filter-list# back ↵
[] spanning-tree# ip-filter-modify/<index> ↵
[] ip-filter-modify/3# info ↵
[] ip-filter-modify/3# status <filterstat> ↵
[] ip-filter-modify/3# control <action> ↵
[] ip-filter-modify/3# interface <port> ↵
[] ip-filter-modify/3# direction <dir> ↵
[] ip-filter-modify/3# broadcast <bc> ↵
[] ip-filter-modify/3# source-ip-address <srcaddr> ↵
[] ip-filter-modify/3# dest-ip-mask <destmask> ↵
[] ip-filter-modify/3# protocol <proto> ↵
[] ip-filter-modify/3# low-source-port <lowsrc> ↵
[] ip-filter-modify/3# high-source-port <hisrc> ↵
[] ip-filter-modify/3# low-dest-port <lowdest> ↵
[] ip-filter-modify/3# high-dest-port <hidest> ↵
```

where	is the…
<index>	number of the filter you want to configure
<filterstat>	filter status; one of the following: • **active** to activate an inactive filter • **delete** to remove a configured filter • **not-in-service** to deactivate a filter but not delete it

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

where	is the…
<action>	action to take on packets that match any IP filter: • **accept** (forward filtered packets) • **discard** (discard filtered packets)
<port>	port to monitor: • **ethernet** • **cable** • **ether-and-cable**
<dir>	direction of the packets that this filter acts upon: **inbound**, **outbound**, or **both**
<bc>	flag that determines whether this filter acts upon broadcast packets: **true** or **false**
<srcaddr>	source IP address that this filter acts upon
<destmask>	subnet mask for destination addresses
<srcipmask>	source IP mask
<destipaddr>	destination IP address
<proto>	protocol type; one of **icmp**, **tcp**, **udp**, or **any** (default is **any**)
<lowsrc>	low port number associated with the source IP address, used to match a range of ports: **0** to **65535** (default is **0**)
<hisrc>	high port number associated with the source IP address, used to match a range of ports: **0** to **65535** (default is **65535**)
<lowdest>	low port number associated with the destination IP address, used to match a range of ports: **0** to **65535** (default is **0**)
<hidest>	high port number associated with the destination IP address, used to match a range of ports: **0** to **65535** (default is **65535**)

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

3 Repeat step 3 for each IP filter you want to configure.

4 To configure the CMTS to send a DU (data unit) packet when the IP filter is activated, use the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# info ↵
[] forwarder# send-du-when-ip-filter <action> ↵
```

where

<action>	is the action to take when the ip filter is used: • **on** (sends du (data unit) packet when there is an ip filter option). • **off** (default; no du packet is sent)

5 Type **exit** to return to the Console> prompt.

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

Configuring IGMPv2 filters

1 Set multicast forwarding (IGMPv2) using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# multicast-forwarding <fwd> ↵
[] forwarder# exit ↵
```

where

<fwd>	is the action to take on multicast packets: • **pass** (forward multicast packets) • **discard** (discard multicast packets)

Note: The command sequence has changed from CMTS version 3.

2 Display the multicast (IGMP) filters using the following commands:

```
Console> manage ↵
[] box# ip-level ↵
[] ip-level# show multicast-list ↵
```

3 Configure the IGMP filter using the following commands:

```
Console> manage ↵
[] box# ip-level ↵
[] ip-level# multicast-modify/<ip-addr>/<interface> ↵
[] multicast-modify/224.0.0.0# info ↵
[] multicast-modify/224.0.0.0# admin-state active ↵
```

where	is the…
<ip-addr>	subnet mask for multicast packets that this filter acts upon; multicast IP addresses are in the range from **224.0.0.0** to **239.255.255.255**
<interface>	physical interface for this filter: **ethernet** or **cable**

4 Type **exit** to return to the Console> prompt.

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

Configuring ICMP Type 9 Router Advertisement Messages filters

1 Set ICMP Type 9 filtering action to enabled or disabled using the following commands:

Console> **manage** ↵

[] box# **forwarder** ↵

[] forwarder# **icmp-type9-filter <fwd>** ↵

where

<fwd>	is the action to take on multicast packets: • **forward** (forward Router Advertisement Messages) • **filter** (discard Router Advertisement Messages)

2 Type **exit** to return to the Console> prompt.

Displaying and configuring port filters

1 The CMTS CLI can display a list of filtered receive port(s) and their corresponding MAC addresses, and whether the receive port has been blocked, or has ethernet and/or cable connectivity. Display the list of port filters using these commands:

Console> **manage** ↵

[] box# **forwarder** ↵

[] forwarder# **show port-filter-list** ↵

The table items are:

- Mac Address of the receive port on the CMTS
- Receive Port mode: none, blocked, ethernet, cable, both
- Allowed-To-Go-To-Port: none, ethernet, cable, both
- Status: other, delete-now, permanent, delete-on-reset, delete-on-timeout

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

2 The port status condition for an individual MAC address can be displayed using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# show port-filter-modify/<mac-addr>/both-ports ↵
```

The table items are:

- Mac Address of the receive port on the CMTS
- Receive Port mode: none, blocked, ethernet, cable, both

3 The port status condition for an individual MAC address for both ports can be modified using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# port-filter-modify/<mac-addr>/both-ports ↵
[] port-filter-modify/<mac-addr>/both-ports# info ↵
[] port-filter-modify/<mac-addr>/both-ports# allowed-to-go-to-port <action>↵
```

where

<action>	defines the receiver port connectivity: • **none** (all connectivity blocked) • **ethernet** (connectivity through the Ethernet port) • **cable** (connectivity through the cable port) • **ethernet+cable** (connectivity through both ports)

```
[] port-filter-modify/<mac-addr>/both-ports# status <action> ↵
```

where

<effect>	defines the effective time of the action: • **other** (default) • **delete now** (filter action is immediate) • **permanent** • **delete-on-reset** • **delete-on-timeout**

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

4 The port status condition for an individual MAC address for the Ethernet port can be modified using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# port-filter-modify/<mac-addr>/ethernet ↵
[] port-filter-modify/<mac-addr>/ethernet# info ↵
[] port-filter-modify/<mac-addr>/ethernet# allowed-to-go-to-port <action>↵
[] port-filter-modify/<mac-addr>/ethernet# status <effect>↵
```

where

<mac-addr> is the MAC address of the Ethernet port

<action> defines the receiver port connectivity:

- **none** (all connectivity blocked)
- **ethernet** (connectivity through the Ethernet port)
- **cable** (connectivity through the cable port)
- **ethernet+cable** (connectivity through both ports)

<effect> defines the effective time of the action:

- **other** (default)
- **delete now** (filter action is immediate)
- **permanent**
- **delete-on-reset**
- **delete-on-timeout**

—continued—

Procedure 5-1 (continued)

Configuring filters using the CLI

5 The port status condition for an individual MAC address for the cable port can be modified using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# port-filter-modify/<mac-addr>/cable ↵
[] port-filter-modify/<mac-addr>/cable# info ↵
[] port-filter-modify/<mac-addr>/cable# allowed-to-go-to-port <action>↵
[] port-filter-modify/<mac-addr>/cable# status <effect>↵
```

where

<mac-addr>	is the MAC address for the cable port
<action>	defines the receiver port connectivity: • **none** (all connectivity blocked) • **ethernet** (connectivity through the Ethernet port) • **cable** (connectivity through the cable port) • **ethernet+cable** (connectivity through both ports)
<effect>	defines the effective time of the action: • **other** (default) • **delete now** (filter action is immediate) • **permanent** • **delete-on-reset** • **delete-on-timeout**

6 Type **exit** to return to the Console> prompt.

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

Configuring Spanning Tree filters

1 Spanning tree filtering is set using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# info ↵
[] forwarder# stpcontrol <action> ↵
```

where

<action>	is the action to take on spanning tree packets: • **st-enabled** (default) enables spanning tree packet forwarding) • **no-st-filter-bpdu** (no spanning tree; bridge protocol data units are filtered) • **no-st-pass-bpdu** (no spanning tree; bridge protocol data units are passed))

2 Type **exit** to return to the Console> prompt.

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

Configuring Proxy ARP filters

1 Set Proxy ARP forwarding using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# proxy-arp <action> ↵
```

where

<action>	is the action that controls the CMTS Proxy ARP in the transparent mode: • **enable** (The CMTS responds to an ARP, received on the Ethernet interface, that is destined for a known device on the HFC network. The CMTS forms a response that appears to be from that device.) • **disable** (Default, where the CMTS does not respond to any ARP requests.)

2 Configure the Proxy ARP cache timeout period using the following commands:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# timeout-proxy-arp <timeout> ↵
```

where

<timeout>	is the timeout (in seconds) for the Proxy ARP cache enable; default is 900 seconds.

3 Display the ARP list for the CMTS using the following commands:

```
Console> manage ↵
[] box# ip-level ↵
[] ip-level# show arp-list ↵
```

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

4 Configure the CMTS Ethernet interface that receives the ARP requests using the following commands:

Console> **manage** ↵

[] box# **ip-level** ↵

[] ip-level# **arp-modify/ethernet/<ip-addr>** ↵

[] arp-modify/ethernet/<ip-addr># **info** ↵

where the settable parameters are**:**

- interface
- ip-address
- mac-address
- media-type

where	is the…
interface	interface type: • **ethernet** • **cable**
ip-address	IP address that receives the ARP requests.
mac-address	MAC address associated with the IP address receiving ARP requests.
media-type	defined media type: • **other** • **invalid** • **dynamic** • **static**

—continued—

Procedure 5-1 (continued)
Configuring filters using the CLI

5 Configure the CMTS cable interface that receives the ARP requests using the following commands:

Console> **manage** ↵

[] box# **ip-level** ↵

[] ip-level# **arp-modify/cable/<ip-addr>** ↵

[] arp-modify/cable/<ip-addr># **info** ↵

where the settable parameters are:

- interface
- ip-address
- mac-address
- media-type

where	is the…
interface	interface type: • **ethernet** • **cable**
ip-address	IP address that receives the ARP requests.
mac-address	MAC address associated with the IP address receiving ARP requests.
media-type	defined media type: • **other** • **invalid** • **dynamic** • **static**

6 Type **exit** to return to the Console> prompt.

—end—

Procedure 5-2
Configuring filters using SNMP

Several MIB tables control packet filtering, as shown in Table 5-1:

Table 5-1
Packet filtering MIB tables

Packet level	MIB table
Link Layer Control (LLC)	docsDevFilterLLCTable
IP	docsDevFilterIpTable
IGMP	lcMulticastGroup
ICMP	lcIpFiltIcmp9
Proxy ARP	lccmtsDPConfiguration

The default actions for LLC and IP filters (when no filters match the packets) can be configured using the following objects.

- docsDevFilterLLCDefault
- docsDevFilterIpDefault
- lcIpFiltIcmp9

Action

Perform the following tasks in this procedure in any order.

Task	Page
Configuring Link Layer Control (LLC) filters	5-23
Configuring IP filters	5-24
Configuring IGMP filters	5-27
Configuring ICMP Type 9 filters	5-28
Configuring Proxy ARP filters	5-29

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

Configuring Link Layer Control (LLC) filters

1 Create or modify an entry in the docsDevFilterLLCTable and set the objects as follows:

Object	**Description**
docsDevFilterLLCIndex	The index number identifying this filter (LLC filter order is irrelevant). You can create up to 10 LLC filters.
docsDevFilterLLCStatus	The status of this filter.
docsDevFilterLLCIFIndex	The interface to be filtered. In cable modems, the default value is the Ethernet interface. For the CMTS, specify one of the following: • **0** — all interfaces • **1** — Ethernet • **2** — Cable (RF)
docsDevFilterLLCProtocolType	The format of the value in docsDevFilterLLCProtocol: • **ethertype** (two-byte Ethernet or SNAP-encapsulated frame) • **dsap** (one-byte 802.2 DSAP value)
docsDevFilterLLCProtocol	A hexadecimal number specifying the LLC protocol to filter. If ProtocolType is **ethertype**, typical values are: • **809B** — Appletalk • **80F3** — Appletalk ARP • **8137** — IPX Ethernet Frame If ProtocolType is **dsap**, typical values are: • **E0** — IPX 802.2/802.3 Frame • **F0** — Netbios Frame

For detailed descriptions of each object, see the *MIB Reference* or the descriptions available through the SNMP manager.

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

2 Set the default action for LLC filtering, by setting the docsDevFilterLLCDefault object as follows:

- **discard** (1)—all packets not matching any LLC filter are discarded
- **accept** (2)—all packets not matching any LLC filter are accepted

3 If you want to configure more filters, return to step 1.

4 Type **exit** to return to the Console> prompt.

Configuring IP filters

1 Create or modify an entry in the docsDevFilterIpTable and set the objects as follows:

Object	Description
docsDevFilterIpIndex	The index used to order the application of filters. The filter with the lowest index is always applied first. You can configure up to 30 IP filters.
docsDevFilterIpStatus	The status of this filter.
docsDevFilterIpControl	Indicates whether packets matching this filter are accepted or discarded.
docsDevFilterIpIfIndex	Indicates the hardware interface that this filter scans: • **0**—the filter applies to all hardware interfaces on the device. • **1**—the Ethernet interface (cable modem) or downstream RF interface (CMTS) • **2**—the RF interface
docsDevFilterIpDirection	Determines whether the filter applies to inbound traffic, outbound traffic, or both.
docsDevFilterIpBroadcast	If set to **true**(1), the filter applies only to multicast and broadcast traffic. If set to **false**(2), the filter applies to all traffic.
docsDevFilterIpSaddr	The source IP address or subnet that is to be matched for this filter.

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

Object	Description
docsDevFilterIpSmask	A bit mask, in IP address format, that is to be applied to the source address before matching. This mask is not necessarily the same as a subnet mask, but uses the same format (1's bits must be leftmost and contiguous).
docsDevFilterIpDaddr	The destination IP address or subnet that is to be matched for this filter.
docsDevFilterIpDmask	A bit mask, in IP address format, that is to be applied to the destination address before matching. This mask is not necessarily the same as a subnet mask, but uses the same format (1's bits must be leftmost and contiguous).
docsDevFilterIpProtocol	The IP protocol (ICMP, TCP, UDP, or Any) that is to be matched.
docsDevFilterIpSourcePortLow	If docsDevFilterIpProtocol is udp or tcp, this is the inclusive lower boundary of the transport-layer source port range that is to be matched.
docsDevFilterIpSourcePortHigh	If docsDevFilterIpProtocol is udp or tcp, this is the inclusive upper boundary of the transport-layer source port range that is to be matched.
docsDevFilterIpDestPortLow	If docsDevFilterIpProtocol is udp or tcp, this is the inclusive lower boundary of the transport-layer destination port range that is to be matched.
docsDevFilterIpDestPortHigh	If docsDevFilterIpProtocol is udp or tcp, this is the inclusive upper boundary of the transport-layer destination port range that is to be matched.

For detailed descriptions of each object, see the *MIB Reference* or the descriptions available through the SNMP manager.

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

2 Set the default action for IP filtering, by setting the docsDevFilterIpDefault object as follows:

- **discard** (1)—all packets not matching any IP filter are discarded
- **accept** (2)—all packets not matching any IP filter are accepted

3 If you want to configure more filters, return to step 1.

4 To configure the CMTS to send a du (data unit) packet when the IP filter is activated, set the lcIpFiltSendDu object to **on**(1) to send a data unit if the packet is filtered, or **off**(2) to not send a packet.

5 Type **exit** to return to the Console> prompt.

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

Configuring IGMP filters

1 On each device you wish to configure, open the entry in the igmpInterfaceTable whose igmpInterfaceIfIndex corresponds to the interface you want to configure. Set the objects as follows (in most cases, you can accept the defaults):

Object	**Description**
igmpInterfaceQueryInterval	The interval, in seconds, at which the device transmits IGMP Host-Query packets on this interface. Default: **125** (seconds)
igmpInterfaceStatus	Set to active to enable IGMP on the interface. Set to destroy to disable IGMP on the interface.
igmpInterfaceVersion	The version of IGMP used on your network. For IGMP to function correctly, all routers on a LAN must be configured to run the same version of IGMP on that LAN. Default: **2**
igmpInterfaceQueryMax-ResponseTime	The maximum query response time, in 1/10 second intervals, advertised in IGMPv2 queries on this interface. Default: **100** (10 seconds)
proxyIfIndex	
igmpInterfaceRobustness	Allows tuning for expected packet loss on a subnet. If you expect a subnet to drop packets, increase the value of this object. IGMP is robust to (Robustness–1) packet losses. Default: **2**

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

Object	Description
igmpInterfaceLastMemb QueryIntvl	The time, in 1/10 second intervals, used as follows: • the maximum response time, inserted into Group-Specific Queries sent in response to Leave Group messages • the amount of time between Group-Specific Query messages Change this value 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 value of this object is irrelevant if igmpInterfaceVersion is version1. Default: **10** (1 second)

2 Under the lcMulticastGroup table, create or modify an entry in the lcMulticastGroupInfo table and set the objects as follows:

Object	Description
lcMulticastForwarding	Specifies action to take for multicast addresses if IGMP is disabled: • **forward**(1) results in all multicasts being forwarded. • **discard**(2) results in all multicasts being discarded. This object has no effect if IGMP is enabled.

Configuring ICMP Type 9 filters

1 Set the lcIpFiltIcmp9 object to configure the ICMP Type 9 filters:

Object	Description
lcIpFiltIcmp9	Specifies action to take for ICMP Type 9 packets on all interfaces: • **on**(1) filters (blocks forwarding) the ICMP Type 9 packets. • **off**(2) forwards the ICMP Type 9 packets.

—continued—

Procedure 5-2 (continued)
Configuring filters using SNMP

Configuring Proxy ARP filters

1 Under the lccmtsDPConfiguration objects, create or modify the following objects:

Object	Description
lcForwardingMode	Controls the Forwarding Data Base Learning Mode: • **none**(1) sets the CMTS to use transparent learning mode. • **dhcp-arp**(2) sets the CMTS to use Layer 3 switching based on DHCP and ARP.
lcProxyArp	Controls proxy ARP on the CMTS in the transparent learning mode: • **enable**(1) makes the CMTS respond to an ARP received from the cable interface when the IP address is known and responds to an ARP received from the Ethernet interface when the IP address is known to be on the cable. • **disable**(2) blocks the CMTS from responding to any ARP messages.
lcProxyArpTimeout	Controls the timeout (in seconds of the Proxy ARP cache entries.
lcArpSpoofingProtection	Prevents any changes to the IP address of a CPE device from its DHCP-assigned value: • **enable** (1) The CMTS does not process ARP packets received from the cable interface, if the sender's address information does not match the internal CMTS FDB. • **disable** (2) The CMTS forwards the ARP packet and updates the ARP cache with the ARP IP address.
lcFunnelMode	• **enable** (1) Funnel mode is enabled. • **disable** (2) Funnel mode is disabled.

For detailed descriptions of each object, see the *MIB Reference* or the descriptions available through the SNMP manager.

—end—

Enhancing network security

This chapter describes the network security features of the CMTS, and how to use them to enhance the security of your broadband network. The following features are available:

- Baseline Privacy (BPI)—encrypts subscriber traffic
- Baseline Privacy Plus (BPI+)—includes the BPI encryption features, plus cable modem authentication
- Subscriber Management—CMTS-based filtering and device limiting (DOCSIS 1.1 devices only)
- Upstream Transmitter Disable (UP-DIS)—disable a cable modem's upstream transmitter from the CMTS

Network security issues

There are two broad categories of network security, in the context of threats within cable data networks:

- Theft of service—individuals with physical access to the network (that is, subscribers or potential subscribers) may attempt to gain access or extended access without authorization. The features described in this chapter inhibit the most common theft of (data) service problems.

Purpose of BPI and BPI+

Purpose of BPI

BPI is a DOCSIS 1.0 feature that provides data encryption standard (DES) encryption of data packets in the cable modem network, preventing eavesdropping on subscriber data connections.

> ***Note:*** BPI data encryption works only between the CMTS and cable modems. Data leaving the cable network is sent unencrypted (unless the subscriber is using SSH or another encryption scheme).

Purpose of BPI+

Baseline Privacy Plus (BPI+) is a DOCSIS 1.1 feature that provides:

- all BPI functionality
- authentication of the cable modems to the CMTS to prevent theft of service

> ***Note:*** BPI+ only provides security services for the cable access network; no security is provided for traffic flows outside this network (for example, past the CMTS or in customer premises equipment).

BPI+ security services

BPI+ provides these security services:

- Media access control (MAC)-layer privacy for upstream and downstream packets through encryption using matched keys.
- Cable modem authentication, which requires each modem to periodically identify itself to the CMTS using an RSA key pair and an X.509 certificate. These security elements are embedded into the cable modem by the manufacturer.
- Access control through a database of cable modems registered and authorized by the CMTS.
- Baseline Privacy Key Management (BPKM), where encryption keys for the different service flows are produced by the CMTS and sent to cable modems to support unicast service flows.

Purpose of Subscriber Management

The C3M (Customer Controlled Cable Modem) uses software running on the subscriber's computer for most functions. A skilled subscriber could thus easily replace the standard cable modem software with software designed to circumvent any imposed filtering or CPE restrictions.

Subscriber Management addresses this potential security issue by duplicating the cable modem filtering and CPE restriction features at the CMTS. You configure Subscriber Management either through a MIB or the CLI.

Subscriber Management is a DOCSIS 1.1 requirement.

Purpose of Upstream Transmitter Disable

This feature is also intended for C3M equipment. It allows network operators to cut off a cable modem (the cable modem must support the UP-DIS message). A cable modem disabled in this manner cannot transmit until the subscriber powers it off then on.

The CMTS 1500 has a proprietary MIB table that contains a list of the MAC addresses of disabled cable modems. During initial ranging, the CMTS examines the table and again sends the UP-DIS command to a cable modem if its MAC address appears in the list.

UP-DIS is a DOCSIS 1.1 requirement for the CMTS, although not all cable modems are required to support it.

Terms and concepts

This chapter uses the following terms and concepts:

C3M
Customer Controlled Cable Modem, a device consisting of (usually) a card that plugs into the subscriber's computer, and cable modem software running on the subscriber's computer.

CPE (or Subscriber host)
Customer Premises Equipment—any subscriber device intending to use the cable data network for communications.

Key
A block of bits used to encrypt or decrypt data on the link.

Service ID (SID)
A number, assigned by the CMTS, to each upstream data link; the SID is used for a number of purposes including BPI.

Certificate
Similar to a key, but used to authenticate cable modems (and similar devices) for use on the network.

Global (network-wide) BPI parameters

Table 6-1 lists global provisionable BPI parameters. These settings are applied to the network, and cover both the CMTS and the cable modems.

Table 6-1
Global BPI parameters

Item and default	Default
Authorization Wait Timeout	10 seconds
Reauthorization Wait Timeout	10 seconds
Authorization Grace Timeout	600 seconds
Operational Wait Timeout	10 seconds
Rekey Wait Timeout	10 seconds
TEK Grace Time	600 seconds
Authorization Reject Wait Timeout	60 seconds
SA Map Wait Timeout	1 second
SA Map Max Retries	4 seconds

BPI initialization sequence

After a cable modem successfully registers on the network, the cable modem and CMTS negotiate encryption as follows:

1 The cable modem sends a public key to the CMTS.

2 CMTS sends an authorization key to the cable modem.

3 The cable modem requests a Traffic Encryption Key (TEK).

4 CMTS sends a Traffic Encryption Key (TEK) back to the cable modem.

Each sequence step can be updated at different intervals.

In this chapter

This chapter contains the procedures listed in Table 6-2.

Table 6-2
Procedures in this chapter

Procedure	Title	Page
6-1	Configuring Baseline Privacy Plus (BPI+) parameters	6-6
6-2	Changing BPI+ certificate information	6-8
6-3	Changing BPI+ authorization for a cable modem	6-10
6-4	Configuring BPI+ Encrypted IP Multicast via SNMP	6-13
6-5	Configuring Subscriber Management	6-16
6-6	Disabling cable modem upstream transmitters	6-24

Procedure 6-1
Configuring Baseline Privacy Plus (BPI+) parameters

Use this procedure to configure BPI+ on your network.

Requirements

To implement BPI+, your cable modems must have an X.509 DER-encoded cable modem certificate.

Action

Perform the tasks in this procedure in any order.

Task	**Page**
Setting BPI+ default lifetime parameters	6-6
Setting timeouts and certification checks	6-7

Setting BPI+ default lifetime parameters

1 Open the docsBpi2CmtsBaseTable and change the following objects:

MIB Object	**Range of Values**	**Default**
docsBPI2CmtsDefaultAuthLifetime	1 to 604800	604800
docsBPI2CmtsDefaultTEKLifetime	1 to 43200	43200

—continued—

Procedure 6-1 (continued)
Configuring Baseline Privacy Plus (BPI+) parameters

Setting timeouts and certification checks

1 Open the docsBpi2CmtsBaseTable and change the following objects:

Object	**Value**
docsBpi2CmtsDefaultSelfSigned-ManufCertTrust	Determines the default trust status of all (new) self-signed manufacturer certificates obtained after setting the object; one of: • **trusted** (1) • **untrusted** (2)
docsBpi2CmtsCheckCertValidityPeriods	Determines whether certificates obtained after setting this object have their validity periods (and their chain's validity periods) checked against the current time of day; one of: • **true** • **false**

—end—

Procedure 6-2
Changing BPI+ certificate information

Use this procedure to override certificate information of a cable modem.

Action

Perform the tasks in this procedure in any order.

Task	**Page**
Overriding certificate information for a cable modem on the CMTS	6-8
Creating or overriding CA (Certificate Authority) certificate information on the CMTS	6-9

Overriding certificate information for a cable modem on the CMTS

1 Open the docsBpi2CmtsProvisionedCmCertTable.

2 Create an entry whose docsBpi2CmtsProvisionedCmCertMacAddress object corresponds to the cable modem’s MAC address, and change the following objects:

Object	**Value**
docsBpi2CmtsProvisionedCmCertTrust	The trust state for the provisioned cable modem certificate entry; one of: • **trusted** (1) • **untrusted** (2)
docsBpi2CmtsProvisionedCmCert	An X509 DER-encoded certificate authority certificate.
docsBpi2CmtsProvisionedCmCertStatus	Set to **Create** (or **wait**, or **go**)

—continued—

Procedure 6-2 (continued)
Changing BPI+ certificate information

Creating or overriding CA (Certificate Authority) certificate information on the CMTS

1 Open the docsBpi2CmtsCACertTable.

2 Create an entry with docsBpi2CmtsCACertIndex 1to10000, and change the following objects:

Object	Value
docsBpi2CmtsCACertTrust	The trust state for the provisioned CMTS certificate entry; one of: • **trusted** (1) • **untrusted** (2) • **chained** (3) • **root** (4)—use only for certificates signed by a root authority
docsBpi2CmtsCACertStatus	Set to **active** or **inactive** as desired or **Create** (and wait or go).
docsBpi2CmtsCACert	An X509 DER-encoded certificate authority certificate.

—end—

Procedure 6-3
Changing BPI+ authorization for a cable modem

Use this procedure to control authorization for individual cable modems.

Action

Perform the tasks in this procedure in any order.

Task	Page
Forcing a cable modem to reauthorize	6-10
Setting and controlling cable modem authorization	6-11
Setting a cable modem Code Verification Certificate (CVC)	6-12

Forcing a cable modem to reauthorize

1 In the network manager, locate and select the cable modem that you want to have reauthorize.

2 Open the docsBpi2CmBaseTable and change the following object:

Object	Value
docsBpi2CmAuthReset	Set to **true** to force the cable modem to reauthorize.

—continued—

Procedure 6-3 (continued)
Changing BPI+ authorization for a cable modem

Setting and controlling cable modem authorization

1 On the CMTS, open the docfsBpi2CmtsAuthTable.

2 Locate the entry whose docsBpi2CmtsAuthCmMacAddress corresponds to the cable modem you want to configure.

3 Change the following objects as desired:

Object	Value
docsBpi2CmtsAuthCmReset	The authorization invalidation action to take; one of the following: • **noResetRequested**(1)—no action • **invalidateAuth**(2)—the CMTS invalidates the current cable modem authorization key(s), but neither transmits an Authorization Invalid message nor invalidates unicast TEKs. • **sendAuthInvalid**(3)—the CMTS invalidates the current cable modem authorization key(s), and transmits an Authorization Invalid message to the cable modem, but does not invalidate unicast TEKs. • **invalidateTeks**(4)—the CMTS invalidates the current cable modem authorization key(s), transmits an Authorization Invalid message to the cable modem, and invalidates all unicast TEKs associated with this cable modem authorization.

—continued—

Procedure 6-3 (continued)
Changing BPI+ authorization for a cable modem

Setting a cable modem Code Verification Certificate (CVC)

1 On the cable modem, open the docsBpi2CodeCvcUpdate object and set its value to the new Code Verification Certificate (CVC).

If the cable modem can upgrade code files, it verifies the CVC and updates the cvcAccessStart value. If the cable modem is not enabled to upgrade code files, it rejects the new CVC.

—end—

Procedure 6-4
Configuring BPI+ Encrypted IP Multicast via SNMP

Use this procedure to configure BPI+ Encrypted IP Multicast via a SNMP Manager.

About BPI+ Encrypted IP Multicast

The CMTS 1500 supports the use of Dynamic Security Associations (SAs) for the encryption of multicast traffic flows. In order for a multicast flow to be encrypted by a Dynamic SA, it is necessary to provision security association into the CMTS. The cable modem must also be provisioned to access that SA.

The CMTS 1500 supports BPI+ Encrypted IP Multicast via two tables in the DOCSIS BPI+ MIB. The tables should be configured in the order below:

- docsBpi2CmtsMulticastAuthTable (Table 1)
- docsBpi2CmtsIpMulticastMapTable (Table 2)

Considerations

It is recommended that the two tables be configured via a SNMP Manager. The docsBpi2CmtsMulticastAuthTable can be configured using the CLI or a SNMP Manager. However, the docsBpi2CmtsIpMulticastMapTable must be configured via SNMP. There is currently no support in the CLI for the docsBpi2CmtsIpMulticastMapTable since this table requires multiple fields to be set.

docsBpi2CmtsMulticastAuthTable

The docsBpi2CmtsMulticastAuthTable maps SAIDs to modems (via their MAC address).

docsBpi2CmtsIpMulticastMapTable

The docsBpi2CmtsIpIPMulticastMapTable maps multicast addresses to Security Association IDs (SAIDs).

Action

Perform the tasks in this procedure.

Task	**Page**
Creating an entry in the docsBpi2CmtsMulticastAuthTable via SNMP	6-14
Creating an entry in the docsBpi2CmtsIpMulticastMapTable via SNMP	6-15

—continued—

Procedure 6-4 (continued)
Configuring BPI+ Encrypted IP Multicast via SNMP

Creating an entry in the docsBpi2CmtsMulticastAuthTable via SNMP

1 Set the following objects in the docsBpi2CmtsMulticastAuthTable:

Object	Description
IFIndex	Must be set to **2** (Sets the encryption across the cable wire)
SAID	Key used to encrypt multicast downstream flow. Range of multicast security associations between 8192 and 16383.
CmMacAddress	The specific CM MAC Address authorized for the multicast downstream flow.
Control	Row status variable. The choices are: • **Active** • **Not-in-service** • **Create-and-go** • **Create-and-wait** • **Destroy** (Only one choice is allowed)

2 Click on the Set All button.

Sets all objects within the table.

—continued—

Procedure 6-4 (continued)
Configuring BPI+ Encrypted IP Multicast via SNMP

Creating an entry in the docsBpi2CmtsIpMulticastMapTable via SNMP

1 Set the following objects in the docsBpi2CmtsIpMulticastMapTable:

Object	Description
IFIndex	Must be set to **2** (Sets the encryption across the cable wire)
Index	1 to10000 (Value used to identify the index)
Address Type	Set to IPV4 (Type of address)
Address	IPV4 address beginning with 224 (EO) to 239 (EF) (octet string)
Mask Type	Set to IPV4 (Type of mask)
Mask	FF FF FF FF (octet string) (Extends the range of addresses to more than one)
SAType	Must be set to Dynamic
DataEncryptAlg	Set to des56CBCMode (recommended) or des40CBCMode)
DataAuthenAlg	Set to None
Map Control	Row status variable. The choices are: • **Active** • **Not-in-service** • **Create-and-go** • **Create-and-wait** • **Destroy** (Only one choice is allowed)

2 Click on the Set All button.

Sets all objects within the table.

—end—

Procedure 6-5
Configuring Subscriber Management

Use this procedure to configure Subscriber Management for cable modems.

About Subscriber Management

Subscriber Management are MIBs stored in the CMTS that provide filtering and CPE limiting capabilities. While cable modems also have these functions, it may be possible for owners of C3M (Customer Controlled Cable Modem) equipment to disable these functions at the modem. Subscriber Management provides a duplicate mechanism that subscribers cannot modify.

The Subscriber Management MIB consists of the following five tables:

docsSubMgtCpeControlEntry

This table controls the subscriber host addresses. The CMTS creates a row in this table for each cable modem registered. You can configure modems to use either subscriber management or the modem-based filtering functions as needed.

docsSubMgtCpeIpEntry

This table lists the subscriber host addresses known to the CMTS. The CMTS creates rows in this table for each cable modem; the cable modem's configuration file may supply the data, or the CMTS can fill in rows during normal operation.

docsSubMgtPktFilterEntry

This table specifies filtering criteria which can be applied to packets destined to or originating from a cable modem. Rows in this table comprise groups of filters; the order of filters within a group is significant. Multiple cable modems can use a filter group.

docsSubMgtTcpUdpFilterEntry

This table supplements the docsSubMgtPktFilterTable, providing optional TCP or UDP port filtering criteria. Only a few modems should require filters in this table. Rows created in this table must correspond to an existing row in the docsSubMgtPktFilterTable.

docsSubMgtCmFilterEntry

This table assigns an ordered group of filters (from the docsSubMgtPktFilter-Table) to a cable modem. The CMTS creates a row in this table for each cable modem during the registration process.

—continued—

docsSubMgtObjectsInfoEntry

This table sets the default values of each cable modem for subscriber management.

Considerations

Subscriber Management has an impact on CMTS performance, since the CMTS has to handle IP filtering chores previously left to each cable modem. Therefore, you should enable Subscriber Management only for cable modems with a strong potential for abuse (such as C3M modems).

IP filters work by AND'ing the filter's address with the filter mask, and the packet's address with the filter mask, and comparing the two. If they are equal, the filter matches. You can match all addresses by using **0.0.0.0** for both the IP address and the mask.

For other general information about IP filters, see Chapter 5.

Action

Perform the tasks in this procedure in any order.

Task	Page
Configuring cable modems for Subscriber Management via the CLI	6-17
Provisioning cable modems for Subscriber Management via SNMP Manager	6-18
Changing Subscriber Management usage after registration	6-18
Configuring IP filters	6-19
Configuring TCP and UDP filters	6-21
Configuring customer premise equipment (CPE) filters via the CLI	6-22

Configuring cable modems for Subscriber Management via the CLI

1 To configure cable modems for Subscriber Management, use the following commands:

[] Console> manage ↵

[] box# **cable-level** ↵

[] cable-level# **sub-filter-downstream-default** ↵

—continued—

Procedure 6-5 (continued)
Configuring Subscriber Management

Provisioning cable modems for Subscriber Management via SNMP Manager

1 In the provisioning server, change the docsSubMgtCpeControlActive MIB as follows:

If you want to…	**Then** set the MIB to…
enable subscriber management	true
disable subscriber management	false

Note: To minimize performance impact, enable Subscriber Management only for cable modems that require it for proper network security.

Changing Subscriber Management usage after registration

1 Select the CMTS that controls the cable modem that you want to change in the network manager.

2 Open the docsSubMgtCpeControlTable.

3 Locate the entry that corresponds to the cable modem that you want to change, and change the docsSubMgtCpeControlActive MIB as follows:

If you want to…	**Then** set the MIB to…
enable subscriber management	true
disable subscriber management	false

—continued—

Configuring IP filters

1 Create a new group, if necessary, in the docsSubMgtPktFilterTable.

2 Select a filter group and create a filter entry. Set the objects as follows:

Object	**Description**
docsSubMgtPktFilterSrcAddrType	The IP address type for source IP addresses: **ipv4** (default) or **ipv6**
docsSubMgtPktFilterSrcAddr	The source IP address. See “Considerations” on page 6-17 for details.
docsSubMgtPktFilterSrcMaskType	The IP address type for the IP source mask: **ipv4** (default) or **ipv6**
docsSubMgtPktFilterSrcMask	The IP mask for source IP addresses. See “Considerations” on page 6-17 for details.
docsSubMgtPktFilterDstAddrType	The IP address type for destination IP addresses: **ipv4** (default) or **ipv6**
docsSubMgtPktFilterDstAddr	The destination IP address. See “Considerations” on page 6-17 for details.
docsSubMgtPktFilterDstMaskType	The IP address type for the IP destination mask: **ipv4** (default) or **ipv6**
docsSubMgtPktFilterDstMask	The IP mask for destination IP addresses. See“Considerations” on page 6-17 for details.

—continued—

Procedure 6-5 (continued)
Configuring Subscriber Management

Object	Description
docsSubMgtPktFilterUlp	The upper level protocol to match. Some typical values include: **6** (TCP), **17** (UDP), and **256** (all protocols). If the protocol matches one of these, the CMTS checks the docsSubMgtPktTcpUdpFilterTable to determine whether more filter checking is required.
docsSubMgtPktFilterTosValue	The TOS value to match. Default: **0**
docsSubMgtPktFilterTosMask	The mask applied against the TosValue and the IP packet's TOS value. Default: 0 (when the TosValue and mask are both zero, the filter does not consider TOS)
docsSubMgtPktFilterAction	The action to take when this filter matches; one of: • **accept** (default; pass the packet) • **drop** (drop the packet)
docsSubMgtPktFilterStatus	Create or destroy the row; one of the following: • **createAndGo**—Creates a new row and makes it available for use. • **createAndWait**—Creates a new row, but does not make it available for uses. • **destroy**—Deletes all of the instances associated with this row.

—continued—

Procedure 6-5 (continued)
Configuring Subscriber Management

Configuring TCP and UDP filters

1 Open the docsSubMgtTcpUdpFilterTable.

2 Select the row in the table that corresponds to the filter in the docsSubMgtPktFilterTable. Change the objects as follows:

Object	**Description**
docsSubMgtTcpUdpSrcPort	The source port to match. The default of 65536 matches any source port number.
docsSubMgtTcpUdpDstPort	The destination port to match. The default of 65536 matches any destination port number.
docsSubMgtTcpFlagValues	The value of the flags. Note that setting any flags not also set in FlagMask is an error. After AND'ing the packet's flags with the FlagMask, the CMTS compares the result with this value. If the values are identical, the flags match. You can specify that certain flags be OFF to match this filter; for example if the FlagMask is {urgent, syn, fin} and the FlagValues is {urgent}, then the syn and fin flags must be OFF to match.
docsSubMgtTcpFlagMask	The flags to check; one or more of: • **urgent**(0) • **ack**(1) • **push**(2) • **reset**(3) • **syn**(4) • **fin**(5)
docsSubMgtTcpUdpStatus	Create or destroy the row; one of the following: • **createAndGo**—Creates a new row and makes it available for use. • **createAndWait**—Creates a new row, but does not make it available for uses. • **destroy**—Deletes all of the instances associated with this row.

—continued—

Procedure 6-5 (continued)
Configuring Subscriber Management

Configuring customer premise equipment (CPE) filters via the CLI

1 Display the CPE address filter list using the following commands:

```
[] Console> manage ↵
[] box# forwarder ↵
[] forwarder# show cpe-addr-filter-list ↵
```

where

<group>	is the CPE filter group number
<index>	is the CPE filter index number
<srcipaddr>	CPE filter for source IP address
<srcipmask>	CPE filter for source IP mask
<dstipaddr>	CPE filter for destination IP address
<dstipmask>	CPE filter for destination IP mask

2 Display the CPE state filter list with the following commands:

```
[] Console> manage ↵
[] box# forwarder ↵
[] forwarder# show cpe-state-filter-list ↵
```

—continued—

Procedure 6-5 (continued)
Configuring Subscriber Management

3 Enter the following commands to set CPE filters:

```
[] Console> manage ↵
[] box# forwarder ↵
[] forwarder# cpe-filter-modify/<group {1-1024}>/<index {1-1024}> ↵
```

where

<group {1-1024}>	is the CPE group number
<index {1-1024}>	is the index value for the CPE filter
<ULP>	is the CPE filter for Upper Level Protocol (ULP) type
<TOS Value>	is the CPE filter for TOS byte value
<TOS Mask>	is the CPE filter for TOS mask
<Action>	is the CPE filter action
<Matches>	is the count of CPE filter matches for the specified entry (Group/Index)
<Status>	is the CPE filter status

Note: The group/index for the CPE address and CPE state filter lists refer to the same filter.

4 Type **exit** to return to the Console> prompt.

—end—

Procedure 6-6
Disabling cable modem upstream transmitters

Use this procedure to enable or disable cable modems that support the UP-DIS (upstream disable) feature.

The CMTS 1500 keeps a list of up to 100 cable modems that are to be disabled whenever they attempt to register.

Action

Perform the tasks in this procedure in any order.

Task	Page
Disabling a cable modem using SNMP	6-24
Disabling a cable modem using the CLI	6-25
Reinstating a cable modem using the SNMP Manager	6-25

Disabling a cable modem using SNMP

1 Open the lcCmtsCmTxDisableEntry.

2 Create a new row in the table and set the objects as follows:

Object	Description
lcCmtsCmTxDisableMacAddress	The MAC address of the cable modem to disable.
lcCmtsCmTxDisableRowStatus	Set to createAndGo. The CMTS sends the UP-DIS message to the cable modem immediately and each time it attempts to register.

—continued—

Procedure 6-6 (continued)
Disabling cable modem upstream transmitters

Disabling a cable modem using the CLI

1 Display the current list of disabled modems (by MAC address) using the following commands:

```
[] Console> manage ↵
box# cable-level ↵
cable-level# show modem-us-disable-list ↵
```

The CMTS displays the MAC address and row status of the modems. The valid row status states are: Active, Not-in-Service, Not-Ready, and Delete.

2 Add a cable modem to the disabled list or change the row status of an existing entry using the following commands:

```
[] cable-level# modem-us-disable-modify/<mac-addr> ↵
[] modem-us-disable-modify/<mac-addr># row-status <value> ↵
```

where	is the…
<macaddr>	MAC address of the cable modem in format 00:00:00:00:00:00
<value>	row status: **active**, **not-in-service**, **not-ready**, **delete**

3 Type **exit** to return to the Console> prompt.

Reinstating a cable modem using the SNMP Manager

1 If you are using a network manager, open the set last page, locate the row corresponding to the modem, and set to **destroy**.

—end—

Using ingress avoidance to improve system reliability

This chapter describes how to configure the CMTS 1500 Ingress Avoidance feature.

Purpose of ingress avoidance

Ingress avoidance lets you provide reliable upstream connections by reacting to noise on upstream channels.

You configure a set of rules to control ingress avoidance. The rules include when a change should occur and what type of actions should take place. You can use three different techniques, in various combinations:

- shift the upstream frequency
- change the upstream bandwidth
- change the modulation profile

Ingress avoidance and spectrum analysis

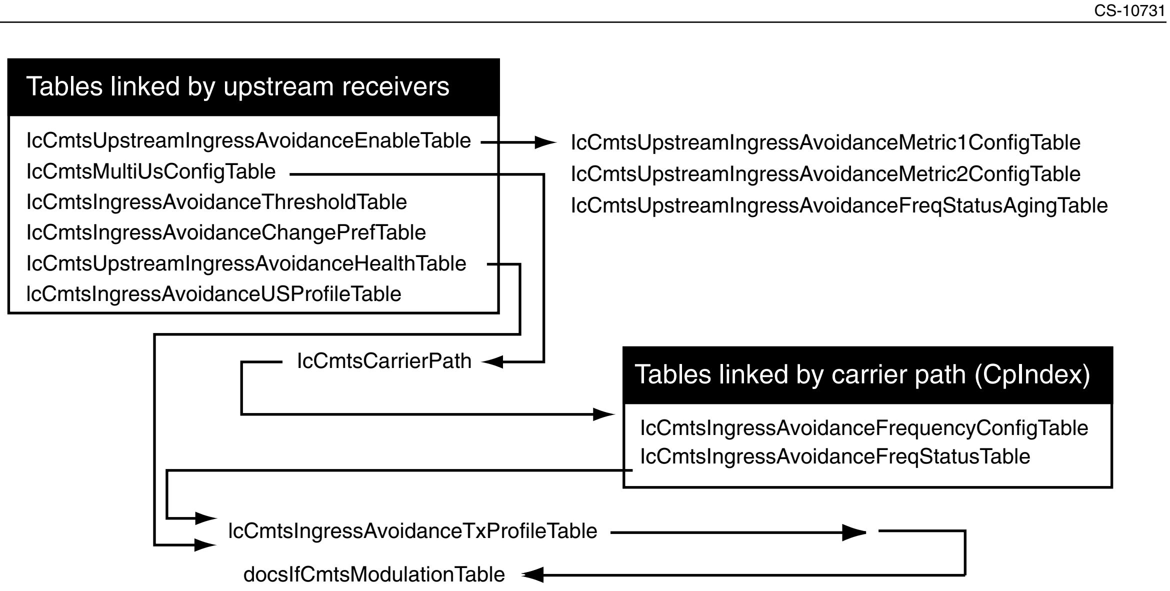
If you configure the eighth upstream receiver as a spectrum analyzer (see Chapter 3), ingress avoidance frequency shift actions choose a frequency with the least amount of noise. Without spectrum analysis, the CMTS can still perform frequency shifts, shifting to the highest available frequency.

Configuration overview

You configure ingress avoidance using 14 MIB tables—13 of these are specific to the Cornerstone CMTS, and one is a standard DOCSIS table. The tables consists of three major groups, and are heavily interrelated. Figure 7-1 shows the relationship among the tables.

Use the procedures in the order printed to configure ingress avoidance.

Figure 7-1
Relationships between ingress avoidance MIBs


CS-10731
Tables linked by upstream receivers
lcCmtsUpstreamIngressAvoidanceEnableTable
lcCmtsMultiUsConfigTable
lcCmtsIngressAvoidanceThresholdTable
lcCmtsIngressAvoidanceChangePrefTable
lcCmtsUpstreamIngressAvoidanceHealthTable
lcCmtsIngressAvoidanceUSProfileTable
lcCmtsUpstreamIngressAvoidanceMetric1ConfigTable
lcCmtsUpstreamIngressAvoidanceMetric2ConfigTable
lcCmtsUpstreamIngressAvoidanceFreqStatusAgingTable
lcCmtsCarrierPath
Tables linked by carrier path (CpIndex)
lcCmtsIngressAvoidanceFrequencyConfigTable
lcCmtsIngressAvoidanceFreqStatusTable
lcCmtsIngressAvoidanceTxProfileTable
docsIfCmtsModulationTable

Terms and concepts

This chapter uses the following terms and concepts:

Ingress
Undesired RF energy entering the cable plant, especially the upstream, from an outside source.

Carrier path
The physical upstream route from a group of cable modems to the CMTS.

Channel group
A group of upstreams with separate physical paths, but share a common upstream frequency and modulation characteristics.

Ingress avoidance action
How the CMTS reacts to ingress beyond a certain level; this can involve shifting the upstream frequency or using lower data rates that can better tolerate the ingress.

Modulation profile
A defined modulation type and forward error correction (a modulation profile can specify other parameters but these are not important for ingress avoidance).

Transmission profile
Consists of a modulation type and bandwidth; ingress avoidance actions work by changing the current avoidance profile.

Metric
A method for determining whether an ingress avoidance action is necessary or imminent.

Procedure 7-1
Preparing to configure ingress avoidance

Use this procedure before configuring ingress avoidance for the first time.

Implementing ingress avoidance requires you to identify upstream receivers that can share a common carrier path or channel group.

CAUTION
Possible loss of functionality
Ingress avoidance functionality requires that upstream receivers sharing the same carrier path be provisioned so their upstream channel operating bandwidth ranges do not overlap.

Each CMTS has a default modulation profile, which you should modify to reflect your upstream plant's baseline performance. For ingress avoidance purposes, you can create modulation profiles that provide more error correction or change the modulation (from QAM16 to QPSK).

Action

Perform the tasks in this procedure in the following order:

Task	Page
Identifying carrier paths and channel groups	7-4
Adding modulation profiles using SNMP	7-5
Adding modulation profiles using the CLI	7-6

Identifying carrier paths and channel groups

1. For each CMTS in your system, inspect the docsIfUpstreamChannelTable MIB object. The entries in this table represent the upstream frequencies configured on the CMTS. Print this table for reference.
2. Determine which upstream frequencies are currently in use by each upstream receiver.
3. Using a plant map or visual inspection and the list from step 1, identify any upstream receivers sharing both:
   - a common return path
   - an upstream frequency

   These receivers may share a carrier path or channel group for ingress avoidance purposes.
4. Repeat steps 1 through 3 for each CMTS in the headend.

—continued—

Procedure 7-1 (continued)
Preparing to configure ingress avoidance

Adding modulation profiles using SNMP

Note: If you do not intend to use modulation profiles with ingress avoidance, you may skip this task.

1 Create an entry in the docsIfCmtsModulationTable and change the following objects:

Object	**Value**
docsIfCmtsModType	The modulation type used on this channel: • **qpsk** • **qam16**
docsIfCmtsModFECErrorCorrection	The number of correctable errored bytes per packet: **0** (FEC not used) to **10**. Higher values prevent retransmission at the expense of some overhead (twice the selected maximum number of correctable bytes).
docsIfCmtsModFECCodewordLength	The number of data bytes (k) in the FEC codeword: **1** to **255**.

Note: This step specifies only those objects that impact ingress avoidance. The other objects have reasonable defaults.

2 Change other objects in this entry, if necessary.

3 Set the docsIfCmtsModControl object to **createAndGo**.

The CMTS saves the entry and makes it available for use.

—continued—

Procedure 7-1 (continued)
Preparing to configure ingress avoidance

Adding modulation profiles using the CLI

Note: If you do not intend to use modulation profiles with ingress avoidance, you may skip this task.

1 Add a modulation profile using the following commands:

```
[] Console> manage ↵
box# cable-level ↵
cable-level# modulation/<index>/<usage> ↵
modulation/1/long-data# info ↵
modulation/1/long-data# type <modtype> ↵
modulation/1/long-data# fec-error-correction <feclen> ↵
modulation/1/long-data# fec-codeword-length <codelen> ↵
```

where	is the…
<index>	modulation profile index: **1** to **10**
<usage>	interval usage type; one of: • **request** • **data-request** • **initial-ranging** • **periodic-ranging** • **short-data** • **long-data**
<modtype>	modulation type used on this channel; one of: • **qpsk** • **qam16**
<feclen>	number of correctable errored bytes per packet: **0** (FEC not used) to **10**. Higher values prevent retransmission at the expense of some overhead (twice the selected maximum number of correctable bytes).
<codelen>	number of data bytes (k) in the FEC codeword: **1** to **255**. Default: **32**

Note: This step specifies only those objects that impact ingress avoidance. The other objects have reasonable defaults.

2 Change other objects in this entry, if necessary.

3 Repeat steps 1 and 2 as needed to set up other modulation profiles.

4 Type **exit** to return to the Console> prompt.

—end—

Procedure 7-2
Setting up carrier paths

Use this procedure to set up carrier paths for ingress avoidance.

A carrier path is the physical route of the upstream path from a group of modems to a CMTS upstream receiver. You can configure up to 10 frequency ranges for use with each carrier path; the CMTS chooses among the configured ranges for frequency shifts.

The CMTS uses these ranges to take ingress avoidance actions by shifting the upstream frequency range (a frequency hop) or changing the transmission profile (which usually reduces the bandwidth or changes modulation type). You can configure ingress avoidance to use only one of these schemes, although using both gives you more flexibility.

CAUTION
Possible loss of functionality
Ingress avoidance functionality requires that upstream receivers sharing the same carrier path be provisioned so their upstream channel operating bandwidth ranges do not overlap.

Note: If you plan to use channel groups (by assigning an upstream receiver to a channel group as shown in “Configuring upstream receiver parameters using SNMP” on page 7-15), all carrier paths within a channel group must be configured identically.

Action

Perform the tasks in this procedure in any order.

Task	Page
Configuring frequency ranges using SNMP	7-8
Configuring frequency ranges using the CLI	7-9

—continued—

Configuring frequency ranges using SNMP

1 Open an entry in the lcCmtsIngressAvoidanceFrequencyConfigTable and change the objects as follows. (Entries are identified by carrier path index and frequency index.)

Object	Value
lcCmtsFreqAvailable	**Yes** to use this frequency range, or **no** to skip this frequency range.
lcCmtsStartFrequency	The frequency (in Hz) of the lower end of the frequency range.
lcCmtsStopFrequency	The frequency (in Hz) of the upper end of the frequency range; this value must be greater than or equal to the start frequency.

Note: You can create frequency ranges in any order. The CMTS does not use the frequency index as a preference, but uses only those frequency ranges with the lcCmtsFreqAvailable object set to **yes**.

2 Repeat step 1 for each frequency range you want to configure. You can configure up to 10 ranges for each carrier path (up to 8 carrier paths).

—continued—

Procedure 7-2 (continued)
Setting up carrier paths

Configuring frequency ranges using the CLI

1 Configure the frequency ranges you want to use using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ingress-avoidance-level ↵
[] ingress-avoidance-level# freq-config/<cp>/<index> ↵
[] freq-config/1/1# info ↵
[] freq-config/1/1# start-frequency <startfreq> ↵
[] freq-config/1/1# stop-frequency <endfreq> ↵
[] freq-config/1/1# freq-available yes ↵
```

where

<cp>	is the carrier path index: **1** to **8**
<index>	is the range index for the designated carrier path: **1** to **10**
<avail>	determines whether to use this frequency range: **yes** or **no**
<startfreq>	is the low frequency in the range you want to test: • **5000000** to **42000000** (5 to 42 MHz) (DOCSIS) • **5000000** to **65000000** (5 to 65 MHz) (EuroDOCSIS)
<endfreq>	is the high frequency in the range you want to test; this value must be greater than or equal to the start frequency: • **5000000** to **42000000** (5 to 42 MHz) (DOCSIS) • **5000000** to **65000000** (5 to 65 MHz) (EuroDOCSIS)

2 Repeat step 2 for each frequency range you want to use.

Note: You cannot change “StartFreq” or “StopFreq” if “Frequency Available” is enabled (“Yes”).

—end—

Procedure 7-3
Configuring ingress avoidance metrics

In ingress avoidance, a **metric** is a method for determining whether an ingress avoidance action is necessary. The CMTS supports two metrics:

Metric	**Description**
Errored Packets	This metric triggers an ingress avoidance action when the ratio of total packets to bad (uncorrectable errored) packets within a given interval falls below a specified threshold. A typical "yellow to red" value is 5000 (if more than one uncorrectable packet occurs per 5000 packets, an ingress avoidance action is necessary). In version 4.0 of CMTS software, a weighting factor provides the ability to take previous data into account; this allows the system to tolerate short-duration error bursts. The formula for calculating the metric is: $\text{metric} = \left(\frac{\text{current \# good packets}}{\text{current total packets}} \times (1 - \text{weight})\right) + (\text{previous metric} \times \text{weight})$ The lcCmtsIngressAvoidanceMetric1Config object controls the behavior of the Errored Packets metric. You can specify that a minimum number of packets must be received during the specified interval before the CMTS takes action on this metric.
Flapping Modem	This is a new metric in version 4.0 of CMTS software. This metric triggers an ingress avoidance action when the percentage of modems that de-register within a given interval exceeds a specified threshold. The lcCmtsIngressAvoidanceMetric2Config object controls the behavior of the Flapping Modem metric. You can specify that a minimum number of cable modems must de-register in the specified interval before the CMTS takes action on this metric.

Aging-out

The ingress avoidance feature has an aging-out function that checks frequencies marked as "bad" and reduces the computed error value until the frequency is again marked as available. All aging is based on a combination of the aging multiplier and the calculation timer. The aging multiplier determines what percentage of the detected ingress noise will be retained after each calculation interval (in milliseconds). The aging multiplier and the calculation timer are configured in the lcCmtsIngressAvoidanceFreqStatusAgingTable.

—continued—

Procedure 7-3 (continued)
Configuring ingress avoidance metrics

Action

Perform the tasks in this procedure in any order.

Task	Page
Configuring ingress avoidance metrics using SNMP	7-11
Configuring ingress avoidance metrics using the CLI	7-13

Configuring ingress avoidance metrics using SNMP

1 Open the lcCmtsIngressAvoidanceMetric1Config table and set the values as follows:

Object	Value
lcCmtsMetric1MinPktsPerSample	The minimum number of packets that must be received during the specified interval. If fewer packets are received, the CMTS does not calculate the metric.
lcCmtsMetric1WeightFactor	The percentage assigned to the previous value for calculating the metric: **0** to **99**. **0** means no weighting towards previous sample. **99** means use only new sample data.
lcCmtsMetric1CalculationTimer	The interval, in milliseconds, between calculations. Minimum value is **100**.

2 Open the lcCmtsIngressAvoidanceMetric2Config table and set the values as follows:

Object	Value
lcCmtsFlappingMinCM	The minimum number of cable modems that must de-register during the specified interval.
lcCmtsFlappingCalculationTimer	The interval, in seconds, between calculations.

—continued—

Procedure 7-3 (continued)
Configuring ingress avoidance metrics

3 Open the entry in the lcCmtsIngressAvoidanceThresholdTable and change the objects as follows:

Object	**Value**
lcCmtsMetric1GreenToYellow	The minimum value of the Errored Packets metric allowed before the CMTS marks the upstream as Yellow.
lcCmtsMetric1YellowToRed	The minimum percentage of the Errored Packets metric allowed before the CMTS performs an ingress avoidance action.
lcCmtsMetric2GreenToYellow	The maximum percentage of the Flapping Modem metric allowed before the CMTS marks the upstream as Yellow.
lcCmtsMetric2YellowToRed	The maximum value of the Flapping Modem metric before the CMTS performs an ingress avoidance action.

Note: The CMTS takes ingress avoidance actions only if ingress avoidance is enabled, regardless of the threshold settings.

—continued—

Procedure 7-3 (continued)
Configuring ingress avoidance metrics

Configuring ingress avoidance metrics using the CLI

1 Configure metric 1 using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ingress-avoidance-level ↵
[] ingress-avoidance-level# metric-config ↵
[] metric-config# info ↵
[] metric-config# min-packets-per-sample <count> ↵
[] metric-config# weight-factor <weight> ↵
[] metric-config# calculation-timer <time> ↵
[] metric-config# back ↵
```

where	is the…
<count>	minimum number of packets that must be received during the specified interval: **1** to **10000000** If fewer packets are received, the CMTS does not calculate the metric.
<weight>	percentage assigned to the previous value for calculating the metric: **0** to **99**
<time>	interval, in milliseconds, between calculations: **50** to **100000000**

2 Configure the metric 1 aging parameters using the following commands:

```
[] ingress-avoidance-level# freq-status-aging-config ↵
[] freq-status-aging-config# info ↵
[] metric1-config# aging-multiplier <percentage> ↵
[] metric1-config# calculation-timer <time> ↵
[] metric1-config# back ↵
```

where	is the…
<percentage>	metric 1 aging multiplier, as a percentage: **1** to **100** This value determines the percentage of the metric to age-out during each timer interval (default is **100**).
<time>	interval, in milliseconds, between aging calculations: **50** to **10000000** (default is 120000 ms)

—continued—

Procedure 7-3 (continued)
Configuring ingress avoidance metrics

3 Configure metric 2 using the following commands:

```
[] ingress-avoidance-level# metric2-config ↵
[] metric2-config# info ↵
[] metric2-config# flapping-min-cms <count> ↵
[] metric2-config# flapping-calculation-timer <time> ↵
[] metric2-config# back ↵
```

where	is the…
<count>	minimum number of cable modems that must be registered to allow metric to be calculated. Minimum is 1; maximum is 10,000
<time>	interval, in seconds, between calculations: **1** to **4294967**

4 Configure the metric thresholds for an upstream, using the following commands:

```
[] ingress-avoidance-level# metric-threshold-config/<ifindex> ↵
[] metric-threshold-config/4# info ↵
[] metric-threshold-config/4# metric-1-green-to-yellow <ratio> ↵
[] metric-threshold-config/4# metric-1-yellow-to-red <ratio> ↵
[] metric-threshold-config/4# metric-2-green-to-yellow <percentage> ↵
[] metric-threshold-config/4# metric-2-yellow-to-red <percentage> ↵
[] metric-threshold-config/4# back ↵
```

where	is the…
<ratio>	ratio of good packets per bad packet When the number of good packets per bad packet falls below this threshold, the CMTS changes the metric 1 status to Yellow or Red as appropriate.
<percentage>	percentage of modems de-registering When the percentage rises above this threshold, the CMTS changes the metric 2 status to Yellow or Red as appropriate.

5 Type **exit** to return to the Console> prompt.

—end—

Procedure 7-4
Configuring upstream receiver parameters

Use this procedure to configure ingress avoidance tables directly related to upstream receivers.

Action

For each upstream receiver installed in the CMTS, perform the tasks in this procedure in any order.

Task	**Page**
Configuring upstream receiver parameters using SNMP	7-15
Configuring upstream receiver parameters using the CLI	7-17
Configuring avoidance and transmission profiles using SNMP	7-19
Configuring avoidance and transmission profiles using the CLI	7-20

Configuring upstream receiver parameters using SNMP

1 Open the entry in the lcCmtsMultiUsConfigTable corresponding to the upstream receiver you want to configure and change the objects as follows:

Object	**Value**
lcCmtsCarrierPath	The carrier path to assign to this upstream receiver: **1** to **8**. ***Note:*** Upstream receivers on the same carrier path cannot operate on the same frequencies.
lcCmtsChannelGroup	The channel group assigned to this receiver, or **0** if no channel group is assigned. ***Note:*** All upstreams in a channel group must operate on the same frequency, modulation profile, and bandwidth.

—continued—

Procedure 7-4 (continued)
Configuring upstream receiver parameters

2 Open the entry in the lcCmtsIngressAvoidanceChangePrefTable corresponding to the upstream receiver you want to configure and change the object as follows:

Object	**Value**
lcCmtsChangePreference	The first change to attempt if ingress avoidance action is necessary; one of: • **profile** • **frequency**

3 Open the entry in the lcCmtsUpstreamIngressAvoidanceEnableTable corresponding to the upstream receiver you want to configure and change the objects as follows:

Object	**Value**
lcCmtsIngressAvoidanceEnable	**On** to enable ingress avoidance for this upstream receiver; **off** to disable ingress avoidance.
lcCmtsMetric1Enable	**On** to enable metric 1 for this upstream receiver; **off** to disable the metric.
lcCmtsMetric2Enable	**On** to enable metric 2 for this upstream receiver; **off** to disable the metric.

Note: You must enable ingress avoidance, **and** one or both metrics, for ingress avoidance to take effect.

4 Repeat steps 1 through 3 for each upstream receiver you are configuring.

—continued—

Procedure 7-4 (continued)
Configuring upstream receiver parameters

Configuring upstream receiver parameters using the CLI

1 Display the current upstream channel information using the following commands:

```
Console> manage ↵
[] box# cable-level ↵
cable-level# show multi-us-list ↵
```

2 Assign a carrier path, and channel group if desired, to an upstream receiver using the following commands:

```
[] box# cable-level ↵
[] cable-level# multi-us-config/<upstream> ↵
[] multi-us-config/4# info ↵
[] multi-us-config/4# carrier-path <cp> ↵
[] multi-us-config/4# channel-group <group> ↵
[] multi-us-config/4# back ↵
```

where	is the…
<upstream>	interface number of the upstream receiver to configure: **4** to **11**
<cp>	carrier path to assign to this upstream receiver: **1** to **8**
<group>	channel group to assign to this upstream receiver: **1** to **8**, or **0** if you are not using channel groups

3 Set the ingress avoidance action preference for an upstream receiver using the following commands:

```
[] cable-level# ingress-avoidance-level ↵
[] ingress-avoidance-level# change-pref/<upstream> ↵
[] change-pref/4# change-preference <pref> ↵
[] change-pref/4# back ↵
```

where	is the…
<upstream>	interface number of the upstream receiver to configure: **4** to **11**
<pref>	first change to attempt if an ingress avoidance action is necessary; one of: • **profile** • **frequency**

—continued—

Procedure 7-4 (continued)
Configuring upstream receiver parameters

4 Enable or disable ingress avoidance and metrics on an upstream receiver using the following commands:

[] ingress-avoidance-level# **enable/<upstream>** ↵

[] enable/4# **info** ↵

[] enable/4# **avoidance-enable <avoidance>** ↵

[] enable/4# **metric1-enable <metric>** ↵

[] enable/4# **metric2-enable <metric>** ↵

[] enable/4# **back** ↵

where	is the…
<upstream>	interface number of the upstream receiver to configure: **4** to **11**
<avoidance>	**On** to enable ingress avoidance for this upstream receiver; **off** to disable ingress avoidance.
<metric>	**On** to enable the selected metric for this upstream receiver; **off** to disable the metric.

Note 1: You must enable ingress avoidance, **and** one or both metrics, for ingress avoidance to take effect.

Note 2: If the eight upstream receiver is configured as a debug spectrum analyzer and Ingress Avoidance (the master control) is enabled without an associated metric enabled, debug spectral analysis does work. Even though the master control is enabled, Ingress Avoidance is NOT enabled unless an associated metric (metric1 or metric2) is also enabled.

5 Repeat steps 1 through 4 for each upstream receiver you want to configure.

6 Type **exit** to return to the Console> prompt.

—continued—

Procedure 7-4 (continued)
Configuring upstream receiver parameters

Configuring avoidance and transmission profiles using SNMP

1 Open an entry in the lcCmtsIngressAvoidanceTxProfileTable and change the objects as follows:

Object	Value
lcCmtsModulationProfileIndex	An entry in the modulation table (value from **1** to **10**). See “Adding modulation profiles using SNMP” on Page 7-5 for more information.

2 Open an entry in the lcCmtsIngressAvoidanceUsProfileTable and change the objects as follows. (Entries are identified by upstream index and preference index.)

Object	Value
lcCmtsStatus	**On** to enable this profile, **off** to disable it.
lcCmtsTransmissionProfileIndex	An entry in the transmission profile table (see step 3).

3 Open an entry in the lcCmtsIngressAvoidanceTxProfileTable and change the objects as follows:

Object	Value
lcCmtsModulationProfileIndex	An entry in the modulation table (value from **1** to **10**).
lcCmtsBandwidth	The bandwidth, in Hz, to use with this transmission profile: **200000**, **400000**, **800000**, **1600000**, or **3200000**.

4 Repeat steps 1 to 3 for each profile you want to configure.

—continued—

Procedure 7-4 (continued)
Configuring upstream receiver parameters

Configuring avoidance and transmission profiles using the CLI

1 Configure an interface and preference combination using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ingress-avoidance-level ↵
[] ingress-avoidance-level# profile-config/<ifIndex>/<pref-index> ↵
[] profile-config/4/1# info ↵
[] profile-config/4/1# status <enable> ↵
[] profile-config/4/1# transmission-profile-index <txprof> ↵
[] profile-config/4/1# back ↵
```

where

<ifIndex>	is the interface index of the upstream receiver you want to configure: **4** to **11** (where **4** is the upstream receiver in slot 1)
<pref-index>	is the index for the ingress avoidance profile: **1** to **3**
<enable>	determines whether to enable this interface and carrier path preference combination: **on** or **off**
<txprof>	is the transmission profile you want to use with this interface and preference combination: **1** to **24**

2 Repeat step 1 for each interface and preference combination you want to configure.

—continued—

Procedure 7-4 (continued)
Configuring upstream receiver parameters

3 Configure a transmission profile using the following commands:

[] ingress-avoidance-level# **tx-profile-config/<index>** ↵

[] tx-profile-config/1# **modulation profile <modindex>**↵

[] tx-profile-config/1# **bandwidth <bw>** ↵

where	is the…
<index>	transmission profile to configure: **1** to **24**
<modindex>	index into the DOCSIS burst profile modulation table
<bw>	bandwidth, in Hz, for this transmission profile: **200000**, **400000**, **800000**, **1600000**, or **3200000**

4 Type **exit** to return to the Console> prompt.

—end—

Procedure 7-5 **Monitoring ingress avoidance performance**

Use this procedure to determine how well ingress avoidance is performing.

Action

Identify the task that corresponds with your preferred interface and follow the steps in that task.

Task	Page
Monitoring ingress avoidance performance using SNMP	7-22
Monitoring ingress avoidance performance using the CLI	7-23

Monitoring ingress avoidance performance using SNMP

1 Proceed as follows:

If you want to monitor…	**Then** go to…
health of the upstream channels	step 2
carrier path frequencies	step 4

2 Access the entry in the lcCmtsUpstreamIngressAvoidanceHealthTable that corresponds to the upstream receiver you want to monitor.

3 Check the lcCmtsMetric1Status and lcCmtsMetric2Status objects.

Both should have a value of **green**.

If either object is **yellow**, monitor the lcCmtsMetric1Value or lcCmtsMetric2Value objects to determine whether the upstream performance is deteriorating.

If either object is **red**, the CMTS is taking an ingress avoidance action. Monitor the lcCmtsHealthProfile and lcCmtsHealthFc objects to determine how the CMTS reacts. The status should return to **green**.

4 Access the entry in the lcCmtsIngressAvoidanceFreqStatusTable that corresponds to the carrier path you want to monitor.

5 Check the lcCmtsFreqStatusAvailable object for any frequency index within the carrier path. Entries marked **inuse** represent the current upstream frequency that the carrier path uses. Entries marked **yes** are not currently being used, but are available for ingress avoidance usage.

6 Check the lcCmtsFreqStatusStatus object for any frequency index within the carrier path. Most frequencies should have a status of **0** (**unk**), which means the frequency is available for use.

Frequencies with a high value in this object are not available. The value should drop over time as the CMTS ages out the status.

—continued—

Procedure 7-5 (continued)
Monitoring ingress avoidance performance

Monitoring ingress avoidance performance using the CLI

1 Enter the following commands to access the ingress avoidance commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ingress-avoidance-level ↵
```

2 Proceed as follows:

If you want to monitor…	**Then** go to…
health of the upstream channels	step 3
carrier path frequencies	step 6

3 Enter the following command to display the frequency health table:

```
[] ingress-avoidance-level# show health-list ↵
```

4 Check the Metric1Status and Metric2Status columns. Frequencies in use should have a value of **green**.

If either metric status is **yellow**, monitor the Metric1Value or Metric2Value columns to determine whether the upstream performance is deteriorating.

If either metric status is **red**, the CMTS is taking an ingress avoidance action. Monitor the Profile and Fc columns to determine how the CMTS reacts. The status should return to **green**.

5 Type **exit** to return to the Console> prompt, or return to step 2 if you want to monitor other parameters.

6 Display the frequency status of any upstream receiver using the following command:

```
[] ingress-avoidance-level# show freq-status-list/<cp> ↵
```

where	is the…
<cp>	carrier path to monitor: **1** to **8**

The CMTS displays the frequencies currently assigned to the selected carrier path, in 200 kHz increments.

7 Check the Available column for any frequency index within the carrier path. Entries marked **inuse** represent the current upstream frequency that the carrier path uses. Entries marked **yes** are not currently being used, but are available for ingress avoidance usage.

—continued—

Procedure 7-5 (continued)
Monitoring ingress avoidance performance

8 Check the Status column for any frequency index within the carrier path. Most frequencies should have a status of **0** (**unk**), which means the frequency is available for use.

Frequencies with a high value in this object are not available. The value should drop over time as the CMTS ages out the status.

9 Enter the following command to display the health status for all channels:

```
[] ingress-avoidance-level# show health-list ↵
```

10 Type **exit** to return to the Console> prompt, or return to step 2 if you want to monitor other parameters.

—end—

Using traps, notifications, and events to monitor system performance

This chapter describes how to configure traps in NMAccess (default mode for the CMTS) and Coexistence modes using the CLI. NMAccess mode supports the SNMPv1 and SNMPv2 protocols using the docsDevNmAccessTable.

Terms and concepts

This chapter uses the following terms and concepts:

Traps

Messages sent when certain events occur, and are generated by SNMPv1 devices. Traps do not have an object id (OID) associated with them, but typically display text messages.

Notifications

Messages used by SNMPv2 manager programs in a manner similar to error messages and warnings. Notifications are actually part of the MIB tree, and have OID values.

Inform

Error and information messages; you can configure the CMTS to report events using SNMP traps or syslog services (or both).

Alarms

Messages specific to the Cornerstone products, to report events not available in the DOCSIS specifications (such as loss of communications).

About traps and notifications

Traps and notifications are messages generated by the CMTS in response to certain system conditions or actions. These messages can be directed to a network management system for display.

SNMPv1 messages are referred to as traps. SNMPv2 messages are referred to as traps (notification) or inform. The SNMPv2 trap (notification) works in a similar fashion to the SNMPv1 trap where a message is sent from the CMTS to the PC. The SNMPv2 inform sends a message from the CMTS to the PC and waits for a reply to be generated from the PC back to the CMTS.

Traps and notification messages can be divided into the following groupings:

- RFC Standard traps:
  - cold start
  - link down
  - link up
  - authentication failure (auth-fail)
- lancity (proprietary)
  - lc-lceventhandler
  - lc-lcderegistration
  - lc-lcredundancy
- Standard DOCSIS traps:
  - RegReq
  - RegRsp
  - RegAck
  - DsxReg
  - DsxRsp
  - DsxAck
  - BPKM
  - Dynsec

Trap support in NMAccess mode

NMAccess mode (default mode for the CMTS) supports the traps listed below via the docsDevNmAccesstable. The trap formats supported in NMAccess mode are SNMPv1 and SNMPv2.

Traps in NMAccess mode are controlled by various MIB objects. For example, the lcTrapTypeTable controls the on/off control for traps one through five, ten, eleven as well as the trap format.

The docsDevEvControlTable controls the on/off control for trap eight, and the lcTrapTypeTable supports the v1/v2 trap formats. The docsDevEvControl and docsDevCmtsTrapControl tables control the on/off control for trap twelve.

> ***Note:*** If the trap is set to On, all entries must be configured correctly in the NMAccess table in order for the trap to work properly.

Trap Number	**Name of Trap**	**Trap Format Supported**
1	coldstart	SNMPv1/v2
3	link-up	SNMPv1/v2
4	link-down	SNMPv1/v2
5	auth-failure	SNMPv1/v2
8	lc-lceventhandler	SNMPv1/v2
10	lc-deregistration	SNMPv1/v2
11	lc-redundancy/failover	SNMPv1/v2
12	standard DOCSIS traps: • RegReq • RegRsp • RegAck • DsxReq • DsxRsp • DsxAck • BpiInit • BPKM • Dynsec	SNMPv1/v2

The trap types for the lcTrapTypeStatus object are:

- send v1 trap (0)
- send v2 notification (1) (default)
- send v2 inform (2)
- no trap (3)

Procedure 8-1
Configuring traps in NMAccess mode using the CLI

This procedure describes how to configure traps in NMAccess mode via the CLI. Traps, which are generated by SNMPv1 devices, typically display text messages when certain events occur.

> *Note:* The CMTS sends traps only to systems that appear in the NmAccessTable with a control setting of **ro-with-traps**, **rw-with-traps**, or **traps-only**.

Requirements

You must be logged into the CMTS CLI with administrative privileges.

Make sure the CMTS is displaying the Console> prompt before starting.

Action

Perform the following tasks in this procedure in any order.

Task	Page
Configuring the default Ethernet link trap action	8-6
Configuring the default cable-level linktrap action	8-6
Configuring the default downstream link trap action	8-7
Configuring the default upstream channel link trap action	8-7
Configuring the default authentication trap action	8-8
Configuring SNMP trap actions	8-9

—continued—

Procedure 8-1 (continued)
Configuring traps and notifications using the CLI

Configuring the default Ethernet link trap action

1 Configure the Ethernet-level link-trap default action using the following commands:

```
Console> manage ↵
[] box# ethernet-level ↵
[] ethernet-level# link-trap <action>↵
```

where

<action>	defines the default setting for the link-trap: • **enabled** (default, traps are passed) • **disabled** (traps are blocked)

2 Type **exit** to return to the Console> prompt.

Note: For traps to be passed, the link-up or link-down trap must be enabled for a particular format.

Configuring the default cable-level linktrap action

1 Configure the cable-level link-trap default action using the following commands:

```
Console> manage ↵
[] box# cable-level ↵
[] cable-level# link-trap <action> ↵
```

where

<action>	defines the default setting for the link-trap: • **enabled** (default, traps are passed) • **disabled** (traps are blocked)

2 Type **exit** to return to the Console> prompt.

Note: For traps to be passed, the link-up or link-down trap must be enabled for a particular format.

—continued—

Procedure 8-1 (continued)
Configuring traps and notifications using the CLI

Configuring the default downstream link trap action

1 Configure the downstream channel link-trap default action using the following commands:

```
Console> manage ↵
[] box# cable-level ↵
[] cable-level# downstream ↵
[] downstream # link-trap <action> ↵
```

where

<action>	defines the default setting for the link-trap: • **enabled** (default, traps are passed) • **disabled** (traps are blocked)

2 Type **exit** to return to the Console> prompt.

Note: For traps to be passed, the link-up or link-down trap must be enabled for a particular format.

Configuring the default upstream channel link trap action

1 Configure the upstream channel link-trap default action for the specified upstream channel using the following commands:

```
Console> manage ↵
[] box# cable-level ↵
[] cable-level# upstream-specific/<channel> ↵
[] upstream-specific/5# link-trap <action> ↵
```

where

<channel>	is the interface index of the upstream channel: **4** through **11**
<action>	defines the default setting for the link-trap: • **enabled** (default, traps are passed) • **disabled** (traps are blocked)

2 Repeat step 1 for each upstream you want to configure.

3 Type **exit** to return to the Console> prompt.

Note: For traps to be passed, the link-up or link-down trap must be enabled for a particular format.

—continued—

Procedure 8-1 (continued)
Configuring traps and notifications using the CLI

Configuring the default authentication trap action

1 Configure the authentication trap default action using the following commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# enable-authen-traps <action> ↵
```

where

<action>	defines the default setting for the link-trap: • **enable** (default, traps are passed) • **disable** (traps are blocked)

2 Type **exit** to return to the Console> prompt.

Note: For traps to be passed, the auth-failure trap must be enabled for a particular format.

—continued—

Procedure 8-1 (continued)
Configuring traps and notifications using the CLI

Configuring SNMP trap actions

1 Configure the SNMP cold start trap default action using the following commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# traps ↵
[] traps# non-docs-traps ↵
[]non-docs-cable-device# info ↵
```

The CMTS displays the non-DOCSIS traps.

2 Set the desired trap action.

3 Use the following commands to configure traps for DOCSIS cable devices:

```
[] traps# docs-traps ↵
[]docs-traps# info ↵
```

The CMTS displays the standard DOCSIS traps.

```
[] docs-cable-device# docsDevCmtsTrapControl <action> ↵
```

where

<action>	defines the status of the trap setting:	
	• (none)	Send notification (default)
	• RegReq	Init Reg Req Fail Trap
	• RegRsp	Init Reg Resp Fail Trap
	• RegAck	Init Reg Ack Fail Trap
	• DsxReg	Dyn Serv Reg Fail Trap
	• DsxRsp	Dyn Serv Resp Fail Trap
	• DsxAck	Dyn Serv Ack Fail Trap
	• BPIInit	Bpi Init Trap
	• BPKM	BPKM Trap
	• Dynsec	Dynamic SA Trap

4 Type **exit** to return to the Console> prompt.

—end—

Procedure 8-2
Configuring traps in NMAccess mode using SNMP

The CMTS traps are controlled through a number of different tables and objects. The following tasks cover the major trap configurations in the CMTS.

Action

Perform the following tasks in this procedure in any order.

Task	**Page**
Configuring the Trap Type Table	8-11
Checking the authentication failure trap messages	8-12

—continued—

Procedure 8-2 (continued)
Configuring traps in NMAccess mode using SNMP

Configuring the Trap Type Table

Note: Event handler traps are controlled through the event handler tables. The lcTrapTypeTable controls the format for event handler traps.

1 Open the lcTrapTypeTable and select one of the following values for lcTrapTypeIndex:

- **coldstartTrap**(1)
- **linkUpTrap**(3)
- **linkDownTrap**(4)
- **authenticationFailureTrap**(5)
- **lc-eventhander**(8)
- **modem-deregistration**(10)
- **redundancy/failover** (11)
- **standard DOCSIS traps** (12)

2 Set the lcTrapTypeStatus object that corresponds to the selected trap type, and set it to one of the following values:

- **send v1 trap**(0)
- **send v2 notification**(1) (default)
- **send v2 inform**(2)
- **no traps**(3)

Note: Option 3 (no traps) is not used for the lc-eventhandler (8) and standard DOCSIS (12) traps. The docsDevEvControlTable controls the on/off control for the lc-eventhandler trap. The docsDevEvControl and docsDevCmtsTrapControl tables control the on/off control for the standard DOCSIS traps.

3 Return to step 1 to set more trap types, if necessary.

—continued—

Procedure 8-2 (continued)
Configuring traps in NMAccess mode using SNMP

Checking the authentication failure trap messages

1 The authentication failure trap messages returned with the trap specify the reason for the failure. These messages are objects as listed in the table below:

Object	Description
lcAuthFailErrorStatus (not-accessible)	Specifies the reason for failure to validate the SNMP request: • **nmAccessTableRestriction**(1) indicates that the requesting network management station does not have appropriate permission to perform the SNMP request. The station IP address may not match an entry in the docsDevNmAccessTable. The community string may not match, or the access privileges were insufficient for the request type. Finally the interface (cable or Ethernet) may not be active. • **noWriteAccessToMibVar**(2) indicates that an SNMP set request was received for a mib variable with a non-write status. • **other**(3) indicates a nonspecific error.
lcAuthFailCommunityString (not accessible)	Shows the community string received in a Get/Set SNMP request that did not pass authentication tests.
lcAuthFailIpAddr (not accessible)	Shows the source IP address of a Get/Set SNMP request that did not pass authentication tests.
lcAuthFailInterface (not accessible)	Shows the interface port on which the Get/Set SNMP request was received, which did not pass authentication tests.

—end—

Procedure 8-3
Configuring CMTS event reporting using the CLI

The CLI commands related to event reporting are found in the **event-level** commands under the **manage** subsystem. These commands configure the following items:

- Syslog server addresses (up to 3)
- Event throttling
- Event routing by severity
- Event viewing

Requirements

You must be logged into the CMTS CLI with administrative privileges.

Make sure the CMTS is displaying the Console> prompt before starting.

Action

Perform the following tasks in this procedure in any order.

Task	**Page**
Setting event log control	8-14
Configuring event throttling	8-14
Configuring the syslog servers	8-15
Handling incoming events	8-16
Clearing the event log	8-17
Viewing the event log	8-17

—continued—

Procedure 8-3 (continued)
Configuring CMTS event reporting using the CLI

Setting event log control

1 Configure the event log control using the following commands:

```
Console> manage ↵
[] box# event-level ↵
[] event-level# control <value> ↵
```

where	
<value>	sets the event log control: **reset-log**, **use-default-reporting**

2 Check the throttle inhibition by entering:

```
Console> manage ↵
[] box# event-level ↵
[] event-level# admin-status-of-throttle <value> ↵
```

3 Type **exit** to return to the Console> prompt.

Configuring event throttling

1 Configure event throttling using the following commands:

```
Console> manage ↵
[] box# event-level ↵
[] event-level# threshold-of-throttle <events> ↵
[] event-level# interval-of-throttle <interval> ↵
```

where	is the…
<events>	maximum number of events that can be received during the throttling interval before the CMTS begins throttling events (default: **100**)
<interval>	interval, in seconds, to count events for throttling purposes (default: **10** seconds)

2 Type **exit** to return to the Console> prompt.

—continued—

Procedure 8-3 (continued)
Configuring CMTS event reporting using the CLI

Configuring the syslog servers

1 Specify up to three syslog servers using the following commands:

Console> **manage** ↵

[] box# **event-level** ↵

[] event-level# **syslog-ip-addr <syslog1>** ↵

[] event-level# **syslog2-ip-addr <syslog2>** ↵

[] event-level# **syslog3-ip-addr <syslog3>** ↵

where

<syslog1> <syslog2> <syslog3>	are the IP addresses of up to three syslog servers that can receive events from the CMTS (the default of **0.0.0.0** disables a server address). The CMTS must be able to communicate with the log server host systems.

2 Type **exit** to return to the Console> prompt.

—continued—

Procedure 8-3 (continued)
Configuring CMTS event reporting using the CLI

Handling incoming events

1 Specify event handling, by severity, using the following commands:

```
Console> manage ↵
[] box# event-level ↵
[] event-level# emergency-reporting <action> ↵
[] event-level# alert-reporting <action> ↵
[] event-level# critical-reporting <action> ↵
[] event-level# error-reporting <action> ↵
[] event-level# warning-reporting <action> ↵
[] event-level# notice-reporting <action> ↵
[] event-level# information-reporting <action> ↵
[] event-level# debug-reporting <action> ↵
```

where

<action> is one or more actions to take upon receiving an event of the specified severity level:

- **none** (default for debug reporting) — ignore events of this level
- **local** (default for all events except debug reporting)— store events of this level in the CMTS event table
- **traps** — generate an SNMP trap for events of this level
- **syslog** — forward events of this level to the configured syslog servers

To specify more than one action, use the plus (+) sign between actions; for example, **local+traps**.

2 Type **exit** to return to the Console> prompt.

—continued—

Procedure 8-3 (continued)
Configuring CMTS event reporting using the CLI

Clearing the event log

1 Clear the CMTS event log using the following commands:

Console> **manage** ↵

[] box# **event-level** ↵

[] event-level# **control reset-log** ↵

2 Type **exit** to return to the Console> prompt.

Viewing the event log

1 Display the CMTS event log using the following commands:

Console> **manage** ↵

[] box# **event-level** ↵

[] event-level# **events-list** ↵

[] event-list# **show** ↵

The CMTS displays the events stored in the event log.

Each displayed event consists of the following items:

- First Date/Time -- The time this entry was created. If this entry represents several identical events, this is the date and time the first event was received.
- Latest Date/Time -- If this entry represents several identical events, this is the date and time the last event was received.
- Repeat Count -- If this entry represents several identical events, this is the number of events represented.
- Event Id -- The event ID number that is generated for each system event.
- Severity Level -- The severity of the event which took place.

2 Type **exit** to return to the Console> prompt.

—end—

Procedure 8-4
Configuring CMTS event reporting using SNMP

The docsDevEvent object contains a variety of controls for recording and throttling traps, events, and log messages. You can configure the following items:

- Syslog server address
- Event throttling
- Event routing by severity

In addition, the docsDevEvent object contains the docsDevEventTable, which contains all recorded events. You can clear this table or view its entries by accessing objects under docsDevEvent.

Action

Perform the following tasks in this procedure in any order.

Task	**Page**
Provisioning event throttling and the Syslog server	8-19
Handling incoming events	8-20
Clearing the event log	8-20

—continued—

Procedure 8-4 (continued)
Configuring CMTS event reporting using SNMP

Provisioning event throttling and the Syslog server

1 Change the following objects in docsDevEventInfo:

Object	**Value**
docsDevEvSyslog	The IP address of the Syslog server, or **0.0.0.0** to disable syslog transmissions.
docsDevEvThrottleAdminStatus	Controls how the CMTS reacts when the trap/syslog threshold is exceeded; one of: • **unconstrained**—ignore the threshold settings • **maintainBelowThreshold**—suppress trap and syslog messages until the next interval • **stopAtThreshold**—stops trap and syslog messages until explicitly restarted (by setting this object again) • **inhibited**—stops all trap and syslog messages
docsDevEvThrottleThreshold	The number of trap or syslog events allowed in the interval before the CMTS throttles transmission. ***Note:***An event causing both a trap and a syslog message is treated as a single event.
docsDevEvThrottleInterval	The interval, in seconds, to count incoming events for throttling.

2 Change the following objects in lcCmtsIf:

Object	**Value**
lcEvSyslog2	The IP address of a second Syslog server, or **0.0.0.0** to disable syslog transmissions.
lcEvSyslog2	The IP address of a third Syslog server, or **0.0.0.0** to disable syslog transmissions.

—continued—

Procedure 8-4 (continued)
Configuring CMTS event reporting using SNMP

Handling incoming events

Note: The docsDevEvControlTable consists of eight entries, ranked by severity. The docsDevEvPriority object acts as an index into this table.

1 Select the severity level by choosing the entry whose docsDevEvPriority object contains the proper value:

- emergency
- alert
- critical
- error
- warning
- notice
- information
- debug

2 Change the docsDevEvReporting object to select one or more actions to take when an event of the selected severity is generated. Specify two or more actions by adding the following values:

Value (hexadecimal)	Action
E0	Local/trap/syslog
C0	Local/trap
A0	Local/syslog
80	Store the events locally
60	Trap/syslog
40	Generate an SNMP trap
20	Generate a syslog message

3 Repeat steps 1 and 2 for each severity level.

Clearing the event log

1 Set the docsDevEvControl object to resetLog.

The CMTS deletes all events from the docsDevEventTable.

—end—

Procedure 8-5
Configuring loss of communications alarms using the CLI

The CMTS supports the loss-of-communications alarm feature found in the ARRIS PacketPort™ and other certain cable modem devices. These alarms alert the operator to network problems, which trigger multiple alarms.

1 Display the number of loss of communications alarms and their states using these commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# nmaccess ↵
[] nmaccess# alarms ↵
[] alarms# show ↵
```

2 Configure the **packet-port-auto-provisioning** parameter to enable or disable auto-provisioning of PacketPort alarms with these commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# nmaccess ↵
[] nmaccess# alarms ↵
[] alarms# packet-port-auto-provisioning <action> ↵
```

where

<action>	determines whether the CMTS allows PacketPort auto-provisioning: • **no** (default) — disable PacketPort autoprovisioning • **yes** — enable PacketPort autoprovisioning ***Note:*** Disabling autoprovisioning does **not** automatically turn off alarm generation as the MIB value is still set.

3 Configure the manager-ip-address parameter to send the alarms to the specified ip address of the alarm manager. using these commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# nmaccess ↵
[] nmaccess# alarms ↵
[] alarms# manager-ip-address <ip-address> ↵
```

—continued—

Procedure 8-5 (continued)
Configuring loss of communications alarms using the CLI

4 Display the list of devices being monitored for loss of communications, and their status, using these commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# nmaccess ↵
[] nmaccess# alarms ↵
[] alarms# active-list ↵
[] active-list# show ↵
```

The CMTS displays the list of active alarms.

5 Display the list of devices with active loss-of-communications alarms using these commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# nmaccess ↵
[] nmaccess# alarms ↵
[] alarms# loc-list ↵
[] loc-list# show ↵
```

The CMTS displays the list of active loss of communication alarms.

6 Configure the loss-of-communications alarm for a device (using the specified MAC address), with these commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# nmaccess ↵
[] nmaccess# alarms ↵
[] alarms# loc-provisioning/<mac-addr> ↵
[] loc-provisioning/<mac-addr># status <status-action> ↵
```

where	is the…
<mac-addr>	MAC address of the interface you want to monitor
<status-action>	action to configure the device alarm; one of: • **active** • **notInService** • **delete**

7 Type **exit** to return to the Console> prompt.

—end—

Procedure 8-6
Configuring loss of communications alarms using SNMP

The CMTS has a set of alarms which alert the operator to loss of communications between the CMTS and devices on the cable network.

Action

Follow these steps to configure loss of communications alarms.

1 Access the IccmtsAlarms objects and view the entries listed in the table below. Configure alarms as desired.

Object	**Description**
IcCmtsCmLossOfCommTotalAlarms (read-only)	Displays the number of devices losing communication with the CMTS.
IcCmtsCmLossOfCommAutoConfigure	Configures the ARRIS PacketPort for loss of communications alarm: **no**(0) **yes**(1)
IcCmtsCmLossOfCommManager	The IP address of the Loss of Communications SNMP manager.

—continued—

Procedure 8-6 (continued)

Configuring loss of communications alarms using SNMP

2 Access the lcCmtsCmLossOfCommAlarmEnableTable and view the entries listed in the table below. Enable alarms as desired.:

Object	**Description**
lcCmtsCmLossOfCommEnableMacAddress (not-accessible)	Displays the number of devices losing communication with the CMTS.
lcCmtsCmLossOfCommEnableStatus (read-create)	Creates or deletes each table entry.
lcCmtsCmLossOfCommEnableUpChannel (read-only)	Shows the upstream channel the device is attached to.
lcCmtsCmLossOfCommEnableCurrentState (read-only)	Indicates the current status of the device: • **initial**(0) • **alarmed**(1) • **alarmSupprd**(2) (alarm suppressed) • **alarmCleared**(3)

—end—

Procedure 8-7
Configuring traps in coexistence mode using a basic configuration

This procedure describes how to configure a basic configuration for the linkup trap in coexistence mode using the CLI.

The following three tables must be configured for basic trap configuration:

- snmpNotifyTable
- snmpTargetAddrTable
- snmpTargetParamTable

The following tables have default settings. The basic configuration given here uses the default settings of the following four tables:

- snmpCommunityTable
- vacmSecurityToGroupTable
- vacmAccessTable
- vacmViewTreeFamilyTable

> ***Note:*** If you use the default settings of the tables listed above, you only need to configure the top three tables for trap configuration. If you do not use the default settings, then you need to configure all nine tables.

To configure the linkup trap using the basic configuration, the following tables must be configured in the order below:

snmpNotifyTable	Table 1
snmpTargetAddrTable	Table 2
snmpTargetParamTable	Table 3

Requirements

You must be logged into the CMTS CLI with administrative privileges.

Make sure the CMTS is displaying the Console> prompt before starting.

—continued—

Trap support in Coexistence mode

Coexistence mode supports the traps listed below. Both SNMPv1 and SNMPv2 trap formats are supported in coexistence mode. In coexistence mode, the docsDevNmAccesstable does not control the traps. The snmpNotifyTable and snmpTargetParamsTable control the following trap formats:

- inform
- notification

The lcTrapTypeTable controls the on/off control for traps one through five, ten, and eleven. The docsDevEvControlTable controls the on/off control for traps eight and eleven.

Trap Number	Name of Trap	Trap Format Supported
1	coldstart	Notification/Inform
3	link-up	Notification/Inform
4	link-down	Notification/Inform
5	auth-failure	Notification/Inform
8	lc-lceventhandler	Notification/Inform
10	lc-deregistration	Notification/Inform
11	lc-redundancy/failover	Notification/Inform
12	standard DOCSIS traps: • RegReq • RegRsp • RegAck • DsxReq • DsxRsp • DsxAck • BpiInit • BPKM • Dynsec	Notification/Inform

Note: If the trap is set to On, all tables must be configured correctly in order for the trap to work properly. Refer to Procedure 8-7, Configuring Traps in Coexistence Mode Using a Basic Configuration, of this module.

—continued—

Procedure 8-7 (continued)

Configuring traps in coexistence mode using a basic configuration

Action

Follow these steps to configure the linkup trap.

1 Configure the linkup trap using the following CLI commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# coex ↵
```

The CMTS SNMP operation is set to coexistence mode.

2 Configure the snmpNotifyTable using the following commands:

```
[] coex# ? ↵
[] coex# show snmpnotify-list ↵
```

The CMTS displays the snmpNotify-List.

Name	Description
NotifyName	Unique identifier used to index a table
Status	Status of entry
NotifyTag	Tag value used to select entries in the snmpTargetAddrTable.
NotifyType	Specifies whether to generate an SNMPv2-Trap PDU or an Inform PDU

```
[] coex# snmpnotify-s/<notifyname> ↵
```

where

<notifyname> is the name of the identifier used to index a table

—continued—

Procedure 8-7 (continued)
Configuring traps in coexistence mode using a basic configuration

[] snmpnotify-specific/notifyname# **info** ↵

The CMTS displays the parameters for snmpnotifyname.

Parameter	**Value**
Notify-tag	Blank (must be filled in otherwise will not point to snmpTargetAddrTable) (Pointer to snmpTargetAddrTable)
Notify-type	• Notification (default) • Inform (Only one choice is allowed) ***Note:*** Trap types default to V2 Notification
Storage-types	nonVolatile (default)
Status	• Active • Not-in-service (default) • Delete (Only one choice is allowed)

[] snmpnotify-specific/notifyname# **status <value>** ↵

where

<value>	is the status of an entry. The choices are: • Active • Not-in-service • Delete (Only one choice is allowed)

[] snmpnotify-specific/notifyname# **notify-tag <value>** ↵

where

<value>	is the indicated value of the notify-tag parameter. This value must be filled in; otherwise this parameter does NOT point to the snmpTargetAddrTable. (Pointer to snmpTargetAddrTable)

[] snmpnotify-specific/notifyname# **back** ↵

The CMTS takes you back to the coexistence command.

—continued—

Procedure 8-7 (continued)
Configuring traps in coexistence mode using a basic configuration

3 Configure the snmpTargetAddrTable using the following commands:

[] coex# **show snmptargetaddr-list** ↵

The CMTS displays the snmptargetaddr-list.

Name	**Description**
TargetName	Unique identifier used to index a table
Status	Status of entry
Target	Domain Specifies the transport type of the address
TargetAddress	Specifies the target address

[] coex# **snmptargetaddr-s/<targetname>** ↵

where

<targetname> is the name of the unique identifier used to index a table

[] snmptargetaddr-specific/targetname# **info** ↵

The CMTS displays the parameters for snmptargetaddr-specific/targetname.

Parameter	**Value**
Addrtdomain	1.3.6.1.6.1.1 (default)
Targetaddress	255.255.255.255/162 (default)
Timeout	1500 100ths-of-seconds (Value from 0 to 2147483647)
Retrycount	3 (Value from 0 to 255)
Taglist	Blank (default)
Parameters	"" (Pointer to snmpTargetParametersTable)
Storage-types	nonVolatile (default)
Status	• Active • Not-in-service (default) • Not-ready • Delete (Only one choice is allowed)

—continued—

Procedure 8-7 (continued)
Configuring traps in coexistence mode using a basic configuration

[] snmptargetaddr-specific/targetname# **taglist <value>** ↵

where

<value>	is the indicated value for the taglist parameter

Note: Note: Per the RFC, the maximum length of characters that make up the taglist is 255. The CLI limits the length of the taglist to approximately 90 characters. However, a workaround exists where a string length of up to 255 characters can be entered for the taglist via a SNMP Manager.

[] snmptargetaddr-specific/targetname# **parameters <value>** ↵

where

<value>	is the indicated value for the field name parameters

[] snmptargetaddr-specific/targetname# **status <value>** ↵

where

<value>	is the status of an entry. The choices are: • Active • Not-in-Service • Delete

[] snmptargetaddr-specific/targetname# **back** ↵

The CMTS takes you back to the coexistence command.

—continued—

Procedure 8-7 (continued)
Configuring traps in coexistence mode using a basic configuration

4 Configure the snmpTargetParamTable using the following commands:

[] coex# **show snmptargetparams-list** ↵

The CMTS displays the snmptargetparams-list.

Name	**Description**
TargetParamsName	Unique identifier used to index a table
Status	Status of entry
MP Model	Message processing model used when generating SNMP messages
Sec-Model	Security model used when generating SNMP messages
Sec-Level	Identifies the security level used when generating SNMP messages

[] coex# **snmptargetparams-s/<targetparamsname>** ↵

where

<targetparamsname>	is the name of the unique identifier used to index a table

—continued—

Procedure 8-7 (continued)
Configuring traps in coexistence mode using a basic configuration

[] snmptargetparams-s/targetparamsname# **info** ↵

The CMTS displays the parameters for targetparamsname.

Parameter	**Value**
MP-model	• SNMPv1 (default) • SNMPv2c • SNMPv2u and SNMPv2 • SNMPv3 (Only one choice is allowed)
Security-model	• SNMPv1 (default) • SNMPv2c • USM • Any (Only one choice is allowed)
Security-name	"" (Pointer to snmpCommunityTable and vacmSecurityToGroupTable)
Security-level	• noAuthNoPriv (default) • authNoPriv • authPriv (Only one choice is allowed)
Storage-types	nonVolatile (default)
Status	• Active • Not-in-service (default) • Not-ready • Delete (Only one choice is allowed)

[] snmptargetparams-specific/targetparamsname# **mp-model <value>** ↵

where

<value>	is the indicated value for the mp-model

—continued—

Procedure 8-7 (continued)

Configuring traps in coexistence mode using a basic configuration

```
[] snmptargetparams-specific/targetparamsname# security-model <value> ↵
```

where

<value>	is the indicated value for the security-model

```
[] snmptargetparams-specific/targetparamsname# security-name <value> ↵
```

where

<value>	is the indicated value for the security-name; this value points to the snmpCommunityTable and the vacmSecurityToGroupTable.

```
[] snmptargetparams-specific/targetparamsname# status <value> ↵
```

where

<value>	is the status of an entry. The choices are: • Active • Not-in-Service • Delete

```
[] snmptargetparams-specific/targetparamsname# back ↵
[] coex# exit ↵
```

Note: Note: The basic configuration for the linkup trap is using the default settings of the snmpCommunityTable. Therefore, the trap community string is public_v2.

—end—

Procedure 8-8
Configuring traps in coexistence mode using notification filters via CLI

This procedure describes how to configure traps in coexistence mode using notification filters via the CLI.

Requirements

You must be logged into the CMTS CLI with administrative privileges.

Make sure the CMTS is displaying the Console> prompt before starting.

Action

Follow these steps to configure traps.

1 Configure a trap using a notification filter via the following CLI commands:

```
Console> manage ↵
[] box# snmp ↵
[] snmp# coex ↵
```

The CMTS SNMP operation is set to coexistence mode.

2 Display the snmpTargetParamsTable using the following commands:

```
[] coex# show snmptargetparams-list ↵
```

The CMTS displays the parameters for the snmpTargetParamsTable.

Name	Description
TargetParamsName	Unique identifier used to index a table
Status	Status of entry
MP Model	Message processing model used when generating SNMP messages
Sec-Model	Security model used when generating SNMP messages
Sec-Level	Identifies the security level used when generating SNMP messages

—continued—

Procedure 8-8 (continued)
Configuring traps in coexistence mode using notification filters via CLI

3 Display the snmpNotifyFilterProfileTable using the following commands:

[] coex# **show snmpfilterprofile-list** ↵

The CMTS displays the parameters for the snmpNotifyFilterProfileTable.

Name	Description
snmpTargetParamsName	Unique identifier used to index a table
Status	Status of an entry
ProfileName	Name of the profile

4 Create a profile name using the following commands:

[] coex# **snmpfilterprofile-specific/<snmptargetparamsname>** ↵

[] snmpfilterprofile-specific/snmptargetparamsnam# **profile-name <value>** ↵

where

<snmptargetparamsname>	is the name of the target parameter or identifier used to index a table
<value>	is the name of the profile to be used

5 Change the status of the profile-name to active using the following command:

[]snmpfilterprofile-specific/snmptargetparamsname# **status <value>** ↵

[]snmpfilterprofile-specific/snmptargetparamsname# **back** ↵

where

<value> is the status of an entry. The choices are:

- Active
- Not-in-service
- Not-ready
- Delete

The CMTS takes you back to the coexistence command.

—continued—

Procedure 8-8 (continued)
Configuring traps in coexistence mode using notification filters via CLI

6 Display the snmpFilterTable using the following command:

```
[] coex# show snmpfilter-list ↵
```

The CMTS displays the parameters for the snmpFilterTable.

Name	Description
ProfileName	Name of the profile
FilterSubtree	A MIB subtree, when combined with the value of the filter mask, is used to select entries in the snmpTargetAddressTable.
Status	Status of an entry
Filtertype	Indicates if a filter subtree is included or excluded from a filter.

7 Create a SNMP filter subtree using the following command:

```
[] coex# snmpfilter-specific/<profilename> <filtersubtree> ↵
```

where

<profilename>	is the name of the profile to be used
<filtersubtree>	is the SNMP matched OID string

8 Create a filter-mask using the following command:

```
[] snmpfilter-specific/profilename/filtersubtree# filter-mask <value> ↵
```

where

<value>	is the legal value for the filter mask; must consist of four decimal values each between 0 and 255 separated by spaces.

9 Change the status of the filter-type to "included":

```
[] snmpfilter-specific/profilename/filtersubtree# filter-type <value> ↵
```

where

<value>	is the status of the filter-type. The choices are: • Included • Excluded

—continued—

Procedure 8-8 (continued)

Configuring traps in coexistence mode using notification filters via CLI

10 Change the status of the SNMP filter to active:

```
[] snmpfilter-specific/profilename/filtersubtree# status <value> ↵
```

where

<value> is the status of the SNMP filter. The choices are:

- Active
- Not-in-service
- Delete

—end—

Class of Service (CoS) and Quality of Service (QoS)

This chapter describes the Class of Service (CoS) and Quality of Service (QoS) functions of the CMTS 1500. These two separate functions are used to create different types of services for customers.

Terms and concepts

Best Effort

A Service Flow type suited for typical interactive data transfers (web browsing, low-priority file transfers, and the like). It does not provide bandwidth guarantees.

Committed Information Rate (CIR)

A downstream service type provided by specifying a minimum reserved traffic rate. The CMTS ensures that the user is provided with that traffic rate.

Class of Service

A DOCSIS 1.0 service that allows you to specify maximum (and minimum) bandwidth values and priorities. See “Class of Service (CoS)” on page 9-3 for details.

Classifier

Rules used to classify packets into a Service Flow. The device compares incoming packets to an ordered list of rules at several protocol levels. Each rule is a row in the **docsQosPktClassTable**.

A matching rule provides a Service Flow ID (SFID). All rules need to match for a packet to match a classifier. Packets that do not match any classifiers are assigned to the default (or primary) Service Flow.

Concatenation

Concatenation is an upstream capability of the cable modem, which allows the cable modem to combine multiple packets into one large packet for upstream transmission.

This capability must be supported in both the CMTS and the cable modem.

Fragmentation

Fragmentation is an upstream capability of the cable modem, that allows the cable modem to split large packets into smaller packets for more efficient use of upstream bandwidth.

This capability must be supported in both the CMTS and the cable modem per Service Flow.

Jitter

The maximum difference in the interval between packets. Certain services, such as telephony, require minimal jitter.

Non-real-time Polling Service (nrtPS)

A service flow type suitable for high-bandwidth, jitter-tolerant services such as file transfer.

Primary Service Flow

The default path for maintenance traffic and any packets not classified onto any other Service Flow. All cable modems have a Primary Upstream Service Flow and a Primary Downstream Service Flow.

Real-time Polling Service (rtPS)

A service flow type suited for services using variable packet sizes at periodic intervals, such as MPEG video.

Service Flow

A unidirectional flow of packets, used to define and shape traffic. See "Service flows" on page 9-5 for details.

Static provisioning

A method for provisioning QoS in which each cable modem reads its QoS definitions

Note: You can provision CoS/QoS only by changing configuration (.MD5) files for the particular cable modem. QoS provisioning is **not** available through the CMTS CLI or SNMP. QoS parameters may be changed through DSx messaging.

Unsolicited Grant Service (UGS)

A Service Flow type suited for services requiring constant bandwidth, such as telephony. The flow is given transmit opportunities at regular intervals without needing to request them.

Unsolicited Grant Service with Activity Detection (UGS/AD)

An upstream service flow type suited for services requiring constant bandwidth, but with intervals of inactivity such as telephony. The flow is given grants at regular intervals unless the upstream is inactive. This allows bandwidth to be released for Best Effort and similar traffic when not required for the higher-priority services.

Class of Service (CoS)

Class of Service is a DOCSIS 1.0 standard, replaced with the Quality of Service (QoS) and Service Flow mechanism in DOCSIS 1.1 devices. However, the CMTS 1500 also supports Class of Service for DOCSIS 1.0 cable modems. Your network can work with both DOCSIS 1.1 and DOCSIS 1.0 devices.

Note: Do not confuse Class of Service with DOCSIS 1.1 Service Classes. The latter are used to provide pre-configured sets of QoS data to Service Flows.

Class of Service under DOCSIS 1.1

The CMTS maintains Class of Service settings separately from the QoS and Service Flow settings. The CMTS maps DOCSIS 1.0 modem configurations to equivalent DOCSIS 1.1 Service Flows.

Class of Service is bidirectional—this is different from Service Flows, which are unidirectional.

Class of Service parameters

Table 9-1 describes the parameters and their corresponding MIB variables (stored in entries under the docsIfQosProfileTable object).

Table 9-1
Class of Service parameters

Parameter	MIB variable
Maximum downstream data rate	docsIfQosProfMaxDownBandwidth
Maximum upstream data rate	docsIfQosProfMaxUpBandwidth
Upstream channel priority (0 to 7; 7 is highest)	docsIfQosProfPriority
Guaranteed minimum upstream data rate	docsIfQosProfGuarUpBandwidth
Maximum transmit burst size (on upstream)	docsIfQosProfMaxTxBurst
Baseline Privacy enabled	docsIfQosProfBaselinePrivacy

How the CMTS maps Class of Service to Quality of Service

The CMTS maps class of service parameters, received from DOCSIS 1.0 modems, to corresponding Quality of Service parameters as part of the modem registration process. The sequence is as follows:

1 The network operator creates a .MD5 configuration file that contains the modem parameters, using a provisioning server. This configuration file is saved to the TFTP file server.

2 During cable modem initialization, the modem loads its configuration (.MD5) file using TFTP.

3 After loading its configuration file, the cable modem registers its class of service parameters to the CMTS.

4 The CMTS converts the DOCSIS 1.0 class of service items into DOCSIS 1.1 Quality of Service parameters.

During normal operation, the CMTS uses these QoS parameters for traffic shaping and prioritizing traffic sent to the cable modem.

Quality of Service (QoS)

Quality of Service (QoS) supports multiple service levels in the network. The V4.2 software is DOCSIS 1.1-compliant, which allows the CMTS to support VoIP applications and lets cable operators provide certain enhanced data and communication services.

The material below presents basic information about the QoS parameters that can be observed by operators at the CMTS.

Static QoS

The version 4.0 release of CMTS 1500 software supports **static QoS** provisioning. Using static QoS, provisioned service flows (from the cable modem's .MD5 configuration file) are automatically admitted and activated.

Effectively, static QoS reserves bandwidth for each provisioned UGS service flow. Best Effort service flows use as much of the remaining bandwidth as needed, subject to data rate and priority constraints.

> ***Note:*** QoS provisioning cannot be changed through the CMTS CLI or SNMP.

Dynamic QoS

The version 4.2 release of CMTS 1500 software supports the ability to create, admit, and activate service flows when they are needed and delete service flows when they are no longer needed. For example, provisioning a UGS service flow does not require reserving bandwidth until the subscriber begins (or receives) a phone call.

Service flows

A Service Flow (SF) is a unidirectional flow of packets through the CMTS. In the CMTS 1500, the Service Flow function is used to define and shape traffic.

Important things to know about Service Flows:

- A Service Flow Identifier (SFID) is created for each Service Flow.
- Classifiers classify packets for particular Service Flows.
- The Primary Service Flow carries maintenance traffic and any packets not matched by classifiers.
- An Admitted Qos Parameter Set exists for Service Flows when a QoS parameter set is admitted.
- An Active Service Flow forwards packets and abides by the ActiveQosParamSet rules.

Typical Best Effort parameters

Some QoS parameters associated with a Best Effort Service Flow include:

- Traffic priority
- IP TOS (Type of Service)
- Maximum traffic rate
- Minimum reserved traffic rate
- Minimum reserved rate packet size

Typical UGS parameters

Some provisionable QoS parameters associated with Unsolicited Grant Service traffic include:

- Unsolicited grant size
- Nominal grant interval
- Tolerated grant jitter
- Grants per interval

QoS support in the CMTS 1500

The CMTS 1500 supports the following QoS aspects:

QoS attribute	Use
Registration of 1.0 or 1.1 modems and devices	Supports QoS objects (Service Flows and Classifiers).
Dynamic Service Addition and Deletion (DSA/DSD)	Creates and deletes Service Flows.
Dynamic Service Change (DSC)	Changes the parameters of an existing Service Flow.
Enhanced Service Flow scheduling services	Supports constant bit rate (CBR) connections for applications (such as VoIP) needing guaranteed bandwidth, jitter tolerance levels, and related attributes using UGS Service Flows.
Provisioned Service Flows	Supports flows provisioned and authorized through the CM registration process.
Fragmentation	Splits large packets into smaller packets to make more efficient use of upstream bandwidth. Supports Multiple Grant mode and Piggyback mode.
Concatenation	Cable modems can pack several PDUs into a single frame for upstream transmission.
Classification	Assigns packets to Service Flows.

Admission control for voice traffic

The CMTS V4.0 permit service operators to limit bandwidth used for voice calls, through these mechanisms:

- Operators can provision each upstream channel with a maximum percentage of available bandwidth that can be used for voice traffic.
- Operators can also specify total maximum voice calls per CMTS and per upstream, using the **max-cbr-flows** command at the **cable-level** or **upstream-specific** level, respectively.

V4.2 CLI support for QoS

Operators can view (but not change) supported QoS MIB objects through the **qos-1.1-level** sub-command in the **manage** command structure.

QoS MIB support

The DOCSIS QoS MIB contains the MIB tables and objects used for defining QoS. There are 6 main tables, listed below.

QoS MIB objects are read-only. The tables are loaded when the CMTS registers managed devices, and uses QoS parameters in the .MD5 files which are loaded into the managed devices during initialization or through DSx messaging.

Supported MIB tables

The docsQosMIBObjects items define the set of tables used for QoS. In the V4.2 release, the following MIB tables are used:

docsQosPktClassTable
Describes the packet classification configured in the cable modem or CMTS (“Classifier” table).

docsQosParamSetTable
Describes set of DOCSIS QoS parameters defined in a managed device. Each entry defines one QoS Parameter Set.

docsQosServiceFlowTable
Describes a set of DOCSIS QoS Service Flows in a managed device.

docsQosServiceFlowStatsTable
Describes statistics associated with the Service Flows in a managed device.

docsQosUpstreamStatsTable
Describes statistics associated with upstream Service Flows.

docsQosCmtsMacToSrvFlowTable
Lists the Service Flow IDs associated with a particular cable modem MAC address. This allows for indexing into other docsQos tables that are indexed by docsQosServiceFlowId and ifIndex.

docsQosDynamicServiceStatsTable
Describes statistics associated with the Dynamic Service Flows in a managed device.

docsQosServiceFlowLogTable
Contains a log of the deleted Service Flows in a managed device.

docsQosPHSTable (10)
Describes payload header suppression (PHS) rules.

Overview of QoS configuration and operation

The general steps to configure QoS are:

- Decide the specific QoS parameters sets, Service Flows, and Classifiers to be configured in the managed devices.
- Create an .MD5 file for each managed device that contains the appropriate QoS parameters. These files have the Service Flows, Classifiers, and other configuration parameters required for QoS.

  Note: Version 4.2 software also supports dynamic addition, modification and deletion of service flows, and associated classifiers and QoS parameters using modem initiated dynamic QoS signaling.

- Load the .MD5 configuration file into the managed device. This is done during initialization (through the TFTP configuration file transfer process), or by resetting the managed device so it reloads the .MD5 configuration file.
- The managed device registers with the CMTS. The managed device transfers its configuration items to the CMTS, which stores these settings (in the various MIB tables and instances of MIB objects).
- Verify successful registration and verify QoS parameters in the CMTS, via:

  — CLI **modem state** command

  — CLI **qos-1.1-level** commands (under **cable-level**)

  — MIB objects in QoS MIB tables (using an SNMP Manager).

- If a cable modem has problems registering, check the event log (see Chapter 10) for possible causes.

Procedure 9-1
Checking QoS service flow parameters

Use this procedure to check QoS parameters, using the CMTS CLI, after the managed device (cable modem, Packet Port™, etc.) has successfully registered.

Requirements

You must be logged into the CMTS CLI.

Make sure the CMTS is displaying the Console> prompt before starting.

Action

Perform the tasks in this procedure in any order. See the *CMTS 1500 CLI Reference Guide*, ARSVD00123, for details on responses for each command.

Task	Page
Displaying the service flows	9-11
Displaying the service flow statistics	9-12
Displaying the upstream statistics	9-12
Displaying the classifiers	9-13
Displaying QoS parameters	9-14
Displaying provisioned parameters for a single QoS parameter set	9-14
Displaying active parameters for a QoS parameter set	9-14
Displaying admitted parameters for a QoS parameter set	9-15
Displaying the dynamic service flows	9-15
Displaying service flow log statistics	9-16
Displaying specific service flow log statistics	9-16
Displaying packet header suppression (PHS)	9-17

—continued—

Procedure 9-1 (continued)
Checking QoS service flow parameters

Displaying the service flows

You can display service flows in several formats.

1 Enter the following commands to display a list of cable modem MAC addresses and associated service flows:

```
Console> manage ↵
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show mac-sf-list ↵
```

2 Enter the following commands to display a list of service flow IDs associated with a specific MAC address:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show sf-per-mac-list/<mac> ↵
```

3 Enter the following commands to display a list of service flow identifiers, service flow directions, primary or secondary tags, and SID (service identifier), if any:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show sf-list ↵
```

4 Enter the following commands to display the service flow characteristics for a specific service flow identifier (sfid):

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show sf-specific/<sfid> ↵
```

5 Type **exit** to return to the Console> prompt.

—continued—

Procedure 9-1 (continued)
Checking QoS service flow parameters

Displaying the service flow statistics

Service flow statistics provide data on service flow transmission.

1 Enter the following commands to view the entire service flow statistics list:

Console> **manage** ↵

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show sf-stats-list** ↵

2 Enter the following commands to display the service flow characteristics for a specific service flow identifier (sfid):

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show sf-specific/<sfid>** ↵

3 Type **exit** to return to the Console> prompt.

Displaying the upstream statistics

Upstream statistics show data for upstream packet traffic.

1 Enter the following commands to display the upstream statistics list by SID:

Console> **manage** ↵

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show upstream-stats-list** ↵

2 Enter the following commands to display upstream statistics for a specific SID:

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show upstream-stats-specific/<sfid>** ↵

3 Type **exit** to return to the Console> prompt.

—continued—

Procedure 9-1 (continued)
Checking QoS service flow parameters

Displaying the classifiers

Classifiers are associated with the service flows and class identifiers.

1 Enter the following commands to display a list of classifiers:

```
Console> manage ↵
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show classifier-list ↵
```

The SFID, direction, priority, state, and number of processed packets for each classifier are displayed in a table format.

2 Enter the following commands to display the classifier data for a specific service flow:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show classifier-per-sf-list/<sfid> ↵
```

3 Enter the following commands to display the provisioning for a specific classifier:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> classifier-specific/<sfid>/<class-id> ↵
[] classifier-specific/3/1> show ↵
```

where

<sfid>	is a service flow identifier
<class-id>	is a classifier to display

The CMTS displays the classifiers associated with the specified service flow and classifier.

4 Type **exit** to return to the Console> prompt.

—continued—

Displaying QoS parameters

1 Enter the following commands to display the list of provisioned QoS parameters:

```
Console> manage ↵
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> qos-params-list ↵
[] classifier-list> show ↵
```

The CMTS displays a list of all provisioned QoS parameters:

2 Type **exit** to return to the Console> prompt.

Displaying provisioned parameters for a single QoS parameter set

1 Enter the following commands to display the provisioning for a specific QoS class:

```
Console> manage ↵
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> qos-params-specific/<sfid>/provisioned ↵
[] qos-params-specific/3/1> show ↵
[] qos-params-specific/3/1> info ↵
```

where

<sfid>	is the service flow identifier value

The CMTS displays the parameters associated with the specified QoS.

2 Type **exit** to return to the Console> prompt.

Displaying active parameters for a QoS parameter set

1 Enter the following commands to display the active parameters for a specific QoS class:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> qos-params-specific/<SFID>/active ↵
[] qos-params-specific/3/1> show ↵
[] qos-params-specific/3/1> info ↵
```

The CMTS displays the QoS parameters for a specified active service flow.

2 Type **exit** to return to the Console> prompt.

—continued—

Procedure 9-1 (continued)
Checking QoS service flow parameters

Displaying admitted parameters for a QoS parameter set

1 Enter the following commands to display the admitted parameters for a specific QoS class:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> qos-params-specific/<SFID>/admitted ↵
[] qos-params-specific/3/1> show ↵
[] qos-params-specific/3/1> info ↵
```

2 The CMTS displays the QoS parameters for a specified admitted service flow.

3 Type **exit** to return to the Console> prompt.

Displaying the dynamic service flows

1 Enter the following commands to display a dynamic service flow statistics list:

```
Console> manage ↵
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show dynamic-service-stats ↵
```

The table displays downstream and upstream information.

2 Use the following commands to display an enhanced list of statistics for the downstream dynamic service flows:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show dynamic-service-specific/downstream ↵
```

3 Use the following commands to display an enhanced list of statistics for the upstream dynamic service flows:

```
[] box> cable-level ↵
[] cable-level> qos-1.1-level ↵
[] qos-1.1-level> show dynamic-service-specific/upstream ↵
```

4 Type **exit** to return to the Console> prompt.

—continued—

Procedure 9-1 (continued)
Checking QoS service flow parameters

Displaying service flow log statistics

1 Enter the following commands to display service flow log statistics:

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **sf-log-list** ↵

[] sf-log-list> **show sf-log-list** ↵

The CMTS displays a list of service flow log statistics.

2 Type exit to return to the Console> prompt.

Displaying specific service flow log statistics

1 Enter the following commands to display specific service flow log statistics:

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **sf-log-specific/<index>** ↵

[] sf-log-list> **show sf-log-specific/<index>** ↵

where

<index>	is the index to entries in the service flow log.

The CMTS displays a list of specific service flow log statistics.

2 Type exit to return to the Console> prompt.

—continued—

Procedure 9-1 (continued)
Checking QoS service flow parameters

Displaying packet header suppression (PHS)

The CMTS supports Packet Header Suppression (PHS) as a mechanism to increase available bandwidth in the network. The packet headers are suppressed at the cable modem and then re-constituted at the CMTS, based on defined rules.

1 Enter the following commands to display a list of PHS entries (service flow identifier, classifier identifier, and the PHS index):

Console> **manage** ↵

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show phs-list** ↵

2 Enter the following commands to display a list of PHS items associated with a service flow:

Console> **manage** ↵

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show phs-per-sf-list/<sfid>** ↵

where

<sfid>	is the service flow identifier

3 Enter the following commands to display a (PHS) list for a specific service flow/classifier combination:

Console> **manage** ↵

[] box> **cable-level** ↵

[] cable-level> **qos-1.1-level** ↵

[] qos-1.1-level> **show phs-sf-specific/<sfid>/<cid>** ↵

where

<cid>	is the classifier identifier

4 Type **exit** to return to the Console> prompt.

—end—

Identifying and correcting problems

This chapter covers basic troubleshooting and diagnostics information for the Cornerstone CMTS.

Diagnostics and power-up self tests

The CMTS 1500 performs a series of diagnostics and tests when the unit is first powered on, to detect hardware and software problems. Some of these tests include CPU check, memory check, fans and temperature status, and so forth. The results are displayed on the front panel, and also can be sent to external monitoring systems.

Syslog servers

The CMTS supports a primary Syslog server, and up to two secondary Syslog servers. The IP addresses for these servers are defined through the CLI, the SNMP manager, or the provisioning server.

Use the logs to track events occurring to the CMTS and cable modems. In addition, you can optionally use a CLI command to display the log. You can check the logs to see if certain events such as ARP spoofing by subscriber CPE equipment occurs. Further, the logs show if CMTS resets occur due to particular events or if the reset was triggered by the watchdog timer or a normal power-off.

Error tracing

The CMTS supports an optional error trace function. The error trace saves the system state in case the CMTS crashes or is powered down, and then displays the saved system state on the console port after you reboot the CMTS. The information can be useful to ARRIS engineering for debugging purposes.

Unless you suspect a bug in the CMTS, leave the error trace function off (the default) since error tracing can impact system performance. To enable error tracing, enter the following command at the Console> prompt:

```
error-trace ON ↵
```

Procedure 10-1
General CMTS issues

The following issues may occur during a CMTS 1500 installation:

- CMTS front panel does not display anything.
- CMTS 1500 will not communicate with a host through the console port.
- CMTS 1500 will not communicate with a host through the Ethernet port.

Action

Find the description in the following task list that matches the problem you are having and follow the instructions in that task. If the problem persists, contact your next level of support.

Task	Page
The front panel display is blank	10-2
No console port communication	10-3
No Ethernet communication	10-3

The front panel display is blank

The CMTS 1500 front panel display should appear whenever the unit is powered on, even if it is not connected to the CATV network or to the Ethernet.

1 Check the ON/OFF switch—make sure the CMTS is powered on.

2 Check the power connection (particularly the plug). Make sure that the unit is receiving power.

3 If you are using a power strip or UPS, check on ON/OFF switch and power connections.

4 If you have a -48 VDC system, make sure that the unit is receiving -48 VDC power.

—continued—

Procedure 10-1 (continued)
General CMTS issues

No console port communication

1 Make sure the CMTS is powered on (the front panel should not be blank).

2 Check the serial link cable. Make sure you are using a null-modem cable between the CMTS and the PC or terminal.

3 Check the port settings at the PC or terminal. If the CMTS is using factory default port settings, they are:

- Data rate = **9600**
- Data Bits = **8**
- Stop Bits = **1**
- Parity = **None**
- Flow Control = **None (Off)**

4 If you have a PC connected to the console port, make sure the terminal emulation program is configured to use the correct serial port.

No Ethernet communication

1 Check the Ethernet connection used on the CMTS 1500. You can configure the CMTS to support the following Ethernet connections:

- MAIN only (with automatic or manual switch-over to the AUX port)
- MAIN and AUX
- AUX only Ethernet connections.

The CMTS polls a “keep-alive” server to determine if the current Ethernet connection is active. This polling server must be accessible to both the MAIN and AUX connections.

2 Check the Ethernet cable. Make sure you use a straight-through cable when connecting to a hub or router, or a cross-over cable when connecting directly to a host system.

3 Check the IP Address and MAC Address for the CMTS 1500, using the front panel display.

If the CMTS does not have an IP address (the front panel display reads **0.0.0.0**), either assign an address manually using a console port connection or make sure the local DHCP server is configured properly.

4 Check the CMTS 1500 front panel Ethernet LED.

If the LED is lit or flickering, the CMTS is recognizing Ethernet traffic. Make sure you are attempting to connect to the right IP address (see step 3), and that you can connect to other devices.

5 Check the Ethernet data rate setting. The CMTS supports 10 or 100 Mbit/sec data rates with auto-sensing; you can also specify a fixed rate.

6 Check the number of remote sessions allowed (the factory default is **5**). If you exceed this number, the CMTS 1500 rejects further connections.

—end—

Procedure 10-2 **General cable plant issues**

The following issues may occur during or after a CMTS 1500 installation:

- Cable modems do not receive downstream RF transmission from the CMTS 1500.
- CMTS 1500 does not receive upstream RF transmission from cable modems

Action

Find the description in the following task list that matches the problem you are having and follow the instructions in that task. If the problem persists, contact your next level of support.

Task	Page
No downstream RF communication to one or more cable modems	10-4
No downstream RF communication to any cable modems	10-5
No upstream RF communication to the CMTS	10-6

No downstream RF communication to one or more cable modems

1 Enter the following command at the CMTS Console> prompt:

Console> **modem** ↵

The CMTS displays a list of connected modems.

If the **modem** command shows no modems, proceed to the task “No downstream RF communication to any cable modems” on page 10-5.

The equivalent MIB is the docsIfCmtsCmStatusTable, which can be checked with the SNMP Manager.

2 Call the subscriber to verify that:

- video reception and other services (if provided) are working properly
- the cable modem is properly connected to its power source and the subscriber’s computer, and is powered on
- the cable modem’s RF port is connected to a tap before any set-top device

If the subscriber is having general cable service problems, check the tap and the surrounding plant for ingress or improper configuration (such as misconfigured, loose, or broken taps).

—continued—

Procedure 10-2 (continued)
General cable plant issues

3 Enter the following commands to determine whether the modem is communicating with the CMTS at all:

```
Console> modem activity <modemaddr> all ↵
Console> output on ↵
```

where

<modemaddr>	is the MAC address of the cable modem to test

The CMTS displays a trace of all communications between itself and the cable modem.

If the CMTS does not display any modem activity after roughly 10 minutes, proceed to step 4.

4 Use the **modem** and **modem activity** commands to determine whether other cable modems nearby are operating properly, and proceed as follows:

If nearby modems are…	**Then**
functioning properly	replace the subscriber's cable modem
not functioning properly	check the cable plant nearby for ingress or improper configuration

No downstream RF communication to any cable modems

1 Make sure the downstream channel is set to the proper frequency for your cable system (the factory default is 0 MHz). Make sure the selected downstream frequency is not being used for broadcast or other services.

You can check the downstream frequency using:

- the front panel
- the CLI **cable-level** commands
- the SNMP docsIfDownstreamChannelTable object

2 Check the admin status for the downstream cable interface; make sure the status is **up**.

3 Check the output level of the CMTS 1500, using a spectrum analyzer. The output level must be within the correct range for cable modems to properly receive (the factory default is 51 dBmV).

Assuming a unity-gain cable plant, the output level should be in the range of -000dBmV to +000dBmV.

4 Double-check the CMTS provisioning in the provisioning server.

5 Inspect the cable plant. Check for ingress and verify that the CMTS downstream signal is being transmitted (use a spectrum analyzer set to the downstream frequency). Correct any problems found.

—continued—

Procedure 10-2 (continued)
General cable plant issues

No upstream RF communication to the CMTS

1 Check the cable modem provisioning in the provisioning server. If any errors are found, correct them. Proceed to step 2 if the modem still does not communicate, or no errors were found.

2 Make sure the cable modem is connected to the proper upstream receiver. Check the connections in the headend to determine which return path the cable modem uses.

3 Check the LLC and IP filter settings for improper configuration, and make any necessary changes.

Filters can block all outgoing RF communication at either the cable modem or the CMTS 1500 output.

4 Check the upstream port statistics in the CLI, using the following command:

```
Console> port upstream <n> ↵
```

where

<n>	is the upstream to test: **1** to **8**

If the listing shows that the upstream has no traffic, assign the cable modem to a different upstream (if possible) and then check the cable plant for ingress or improper configuration.

5 Enter the following commands to determine whether the modem is communicating with the CMTS at all:

```
Console> modem activity <modemaddr> all ↵
Console> output on ↵
```

where

<modemaddr>	is the MAC address of the cable modem to test

If the CMTS is receiving responses from the cable modem, the upstream path is working. Check the traffic between the CMTS and cable modem for registration and DHCP problems. Otherwise, proceed to step 6.

—continued—

Procedure 10-2 (continued)
General cable plant issues

6 Call the subscriber to verify that:

- video reception and other services (if provided) are working properly—in particular, services requiring use of the upstream
- the cable modem is properly connected to its power source and the subscriber’s computer, and is powered on
- the cable modem’s RF port is connected to a tap before any set-top device

If the subscriber is having general cable service problems, check the tap and the surrounding plant for ingress or improper configuration (such as misconfigured, loose, or broken taps).

7 Ask the subscriber to inspect the cable modem to verify that it is receiving downstream signals from the CMTS 1500. Some cable modems have an LED activity indicator that flickers when it detects downstream data; others require dispatching a technician to the subscriber site to check the downstream directly.

8 Use the **modem** and **modem activity** commands to determine whether other cable modems nearby are operating properly, and proceed as follows:

If nearby modems are…	**Then**
functioning properly	replace the subscriber’s cable modem
not functioning properly	check the cable plant nearby for ingress or improper configuration

—end—

Procedure 10-3 **Cable modem ranging/registering issues**

The following issues may occur when cable modems attempt to register on the network:

- Cable modems begin ranging but then receive an abort message
- Cable modems cannot obtain an IP address
- The CMTS generates Maps and UCDs, but the modem does not range properly
- A cable modem appears to register, but then reboots or rescans

Action

Find the description in the following task list that matches the problem you are having and follow the instructions in that task. If the problem persists, contact your next level of support.

Task	Page
Modems begin ranging then receive an abort message	10-8
Modem cannot obtain an IP address	10-8
Modem does not range properly	10-9
Modem appears to register but then reboots or rescans	10-9
CMTS is not generating UCDs or maps	10-10

Modems begin ranging then receive an abort message

1 Log onto the provisioning server.

2 In the cable modem provisioning, make sure there is an entry that corresponds to the cable modem’s MAC address.

3 Create or modify provisioning entries as needed.

Modem cannot obtain an IP address

1 Check the DHCP server and make sure that it is operating.

2 In the provisioning server, inspect the IP and LLC filters to make sure that the modem is not blocked.

3 Determine which upstream receiver the cable modem is using, and check other modems on the same upstream.

If all modems on the upstream are having the same problems, use either the CLI **upstream-failover** command or the SNMP lcCmtsRedundancyFailover MIB to switch these modems to the redundant upstream. See the *CMTS 1500 User Guide*, ARSVD00122, for detailed instructions.

—continued—

Procedure 10-3 (continued)
Cable modem ranging/registering issues

Modem does not range properly

1 Enter the following CLI commands to determine whether the modem is receiving initial ranging information:

Console> **modem activity <modemaddr> initial_ranging** ↵

Console> **output on** ↵

where

<modemaddr>	is the MAC address of the cable modem to test

The CMTS displays initial ranging data for the specified modem as it communicates with the cable modem.

2 Observe the console display for several minutes, watching for “Initial Rng Req Rcvd” messages, and then proceed as follows:

If the modem is…	**Then**
requesting ranging data	go to step 3
not requesting ranging data	proceed to “CMTS is not generating UCDs or maps” on page 10-10

3 In the provisioning server, inspect the upstream configuration. Make sure the upstream is configured properly and make corrections as necessary.

Modem appears to register but then reboots or rescans

1 Open the provisioning server’s cable modem configuration entries and inspect the downstream frequency for that modem.

2 Use the **cable-level** and **downstream info** CLI commands, or the SNMP docsIfDownstreamChannelTable object, to check the actual downstream frequency of the cable modem.

3 Proceed as follows:

If the frequencies…	**Then**
match	call your next level of support
do not match	change or delete the frequency in the cable modem provisioning record

—continued—

Procedure 10-3 (continued)
Cable modem ranging/registering issues

CMTS is not generating UCDs or maps

1. Using a network manager, examine the lcCmtsUpstreamTable to determine whether the CMTS has transmitted any maps.
2. Proceed as follows:

If the map counters…	**Then**
have any non-zero values	refer to one of the other tasks in this procedure, since the CMTS is generating maps
all have zero values	proceed to step 3

3. Verify that:
   - the configuration files for the CMTS are configured correctly
   - the CMTS has downloaded the right configuration files from the TFTP server

—end—

Procedure 10-4
Miscellaneous issues

The following issues may occur during normal operation:

- MIB information is missing or incomplete

Action

Find the description in the following task list that matches the problem you are having and follow the instructions in that task. If the problem persists, contact your next level of support.

Task	Page
SNMP manager MIB information is missing or incomplete	10-11

SNMP manager MIB information is missing or incomplete

1 Verify that the correct MIB files are available to the network manager.

2 Make sure the MIB files are loading in the proper order.

3 Reload the MIBs.

—end—

Procedure 10-5
Viewing event tables

The docsDevEventTable contains a list of network and device events that may help to isolate and solve various problems in the network. The docsDevEvControlTable controls whether the CMTS stores events of each severity level in the table (under normal conditions; you may choose to ignore **information** and **debug** events, and perhaps higher-level events). See "Handling incoming events" on page 8-20 for more information.

The docsDevEventTable consists of a series of entries; each entry contains the objects listed in Table 10-1

Table 10-1
Objects in the DocsDevEventEntry object

Object	Description
docsDevEvIndex	The entry's index.
docsDevEvFirstTime	The time this entry was created. If this entry represents several identical events, this is the date and time the first event was received. ***Note:*** The date and time shown may not be accurate.
docsDevEvLastTime	If this entry represents several identical events, this is the date and time the last event was received.
docsDevEvCount	If this entry represents several identical events, this is the number of events represented.
docsDevEvLevel	The severity of the event; one of: • **emergency** • **alert** • **critical** • **error** • **warning** • **notice** • **information** • **debug**
docsDevEvId	Uniquely identifies the event reported by this entry.
docsDevEvText	A text description of the event.

—continued—

Procedure 10-5 (continued)
Viewing event tables

Action

1 Access the docsDevEventTable and view the entries.

2 Take note of higher-level events, especially those with a priority level of **emergency**, **alert**, and **critical**. These events are likely to represent problems affecting service.

—end—

Procedure 10-6
Using the front panel test port

You can use the upstream RF test port, on the front panel of the CMTS 1500, to test the signal entering any upstream receiver. The test port is a standard F-connector.

Restrictions

You cannot use the front panel test port if the eighth upstream receiver is in failover mode. This can happen under the following conditions:

- You have configured the eighth upstream receiver for redundant operation, the CMTS has detected a failure on one of the other upstream receivers, and the CMTS has performed an automatic protection switch.
- You have manually switched the eighth upstream receiver to another upstream using the **upstream-failover** command. To remove the protection switch, use the **upstream-failover 0** command.

If you have configured the eighth upstream receiver as a spectrum analyzer, the CMTS redirects RF to the front panel test port for five minutes before automatically returning to spectrum analysis.

> ***Note:*** For information on configuring the eighth upstream receiver, see Chapter 3.

—continued—

Procedure 10-6 (continued)
Using the front panel test port

Action

Identify the task that corresponds with your preferred interface and follow the steps in that task.

Task	**Page**
Redirecting the front panel test port using the CLI	10-15
Redirecting the front panel test port using SNMP	10-15
Redirecting the front panel test port using the front panel	10-16

Redirecting the front panel test port using the CLI

1 Connect your test equipment to the test port and set it to the upstream frequency you want to test. See the documentation for your test equipment for details.

2 Redirect an upstream channel to the front panel test port using the following commands:

[] Console> **manage** ↵

[] box# **cable-level** ↵

[] cable-level# **upstream-test-port** ↵

[] upstream-test-port# **upstream <channel>** ↵

where

<channel>	is the number of the upstream to redirect (**1** to **8**), or **none** to cancel a previous redirection

Redirecting the front panel test port using SNMP

1 Connect your test equipment to the test port and set it to the upstream frequency you want to test. See the documentation for your test equipment for details.

2 Open the lcCmtsIfInfo group and set the following object:

Object	**Value**
lcCmtsUsTestPort	the number of the upstream to redirect (**1** to **8**), or **none** to cancel a previous redirection

—continued—

Procedure 10-6 (continued)
Using the front panel test port

Redirecting the front panel test port using the front panel

1 Connect your test equipment to the test port and set it to the upstream frequency you want to test. See the documentation for your test equipment for details.

2 Press **Mode** on the front panel until the display reads "`Cornerstone`."

3 Press **Down** on the front panel until the display reads "`Upstream x menu`," where `x` is the upstream you want to redirect.

4 Press **Select**.

5 Press **Down** until the front panel displays "`UpStm x Test Port`."

6 Press **Select** to choose an upstream to redirect.

7 Press **Down** until the front panel displays the upstream receiver test port that you want to view.

8 Press **Select** to redirect the upstream receiver currently displayed to the test port.

—end—

Procedure 10-7
Using the spectrum analyzer for troubleshooting

You can provision the eighth upstream receiver for use as a spectrum analyzer (see Chapter 3 for details). The spectrum analyzer feature is primarily used with ingress avoidance; it automatically monitors frequencies available for use in case the CMTS needs to relocate an upstream carrier path.

When you are not using ingress avoidance, you can manually control the spectrum analyzer to monitor selected frequencies. You may want to do this to locate problem areas in your upstream paths, or to determine how upgrades or changes affect the noise levels in your plant.

Using the spectrum analyzer in this fashion is called “manual” or “debug” mode.

Overview

You use the spectrum analyzer in debug mode as follows:

- Provision the eighth upstream receiver as “spectrum analyzer debug only.”
- Configure frequency ranges to test, using the **freq-config** CLI command or the LcCmtsIngressAvoidanceFrequencyConfig table.
- Display results using the **freq-status-list** CLI command or the LcCmtsIngressAvoidanceFreqStatus table.

Action

Identify the task that corresponds with your preferred interface and follow the steps in that task.

Task	Page
Using the debug spectrum analyzer from the CLI	10-18
Using the debug spectrum analyzer from SNMP	10-19

—continued—

Procedure 10-7 (continued)
Using the spectrum analyzer for troubleshooting

Using the debug spectrum analyzer from the CLI

1. Provision the eighth upstream receiver as **spec-analysis-debug-only** (see “Configuring the eighth upstream receiver using the CLI” on page 3-13).
2. Configure the frequency ranges you want to test using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ingress-avoidance-level ↵
[] ingress-avoidance-level# freq-config/<cp>/<index> ↵
[] freq-config/1/1# info ↵
[] freq-config/1/1# start-frequency <startfreq> ↵
[] freq-config/1/1# stop-frequency <endfreq> ↵
[] freq-config/1/1# freq-available yes ↵
```

where	is the…
<cp>	carrier path index: **1** to **8**
<index>	range index for the designated carrier path: **1** to **10**
<startfreq>	low frequency in the range you want to test: **5000000** to **42000000** (5 to 42 MHz, DOCSIS) or **5000000** to **65000000** (5 to 65 MHz, EuroDOCSIS)
<endfreq>	low frequency in the range you want to test: **5000000** to **42000000** (5 to 42 MHz, DOCSIS) or **5000000** to **65000000** (5 to 65 MHz, EuroDOCSIS) This value must be greater than or equal to the start frequency

3. Repeat step 2 for each frequency range you want to monitor.

—continued—

Procedure 10-7 (continued)
Using the spectrum analyzer for troubleshooting

4 View the results of the spectrum analysis using the following commands:

```
[] Console> manage ↵
[] box# cable-level ↵
[] cable-level# ingress-avoidance-level ↵
[] ingress-avoidance-level# freq-status-list/<cp> ↵
```

where	is the…
<cp>	carrier path index: **1** to **8**

The CMTS displays the status of each frequency in the range, at 200kHz intervals.

Note: The CMTS does not display frequency ranges that are currently in use by the upstream channel.

Using the debug spectrum analyzer from SNMP

1 Change the value of the lcCmtsMultiUsageUsAdmin object to **SpectrumAnalysisDebugOnly** (see “Configuring the eighth upstream receiver using SNMP” on page 3-17).

2 Open an entry in the lcCmtsIngressAvoidanceFrequencyConfigTable and change the objects as follows. (Entries are identified by carrier path index and frequency index.)

Object	**Value**
lcCmtsFreqAvailable	**Yes** to use this frequency range, or **no** to ignore this frequency range.
lcCmtsStartFrequency	The frequency (in Hz) of the lower end of the frequency range.
lcCmtsStopFrequency	The frequency (in Hz) of the upper end of the frequency range.

3 Repeat step 2 for each frequency range you want to configure. You can configure up to 10 ranges for each carrier path (up to 8 carrier paths).

4 View the results of the spectrum analysis using the LcCmtsIngressAvoidanceFreqStatus table. This table displays the status of each frequency in the selected ranges, at 200 kHz intervals.

—end—

Procedure 10-8
Determining upstream bandwidth usage

Use the upstream-bandwidth CLI command to:

- display current upstream bandwidth usage
- calculate the number of UGS service flows that can be reserved on the upstream

Action

Perform any of the following tasks as desired.

Task	Page
Displaying upstream bandwidth usage	10-20
Calculating supported UGS service flows	10-21

Displaying upstream bandwidth usage

1 Display the upstream bandwidth usage using the following command:

[] Console> **upstream-bandwidth display** ↵

The CMTS displays the current usage for each upstream. The following example shows only one of the upstreams.

```
Channel 1: width 1600 KHz (2650000 bits/sec)
           shortest interval scheduled = 20 msecs (51200 bits/6400 bytes)
           48 percent reserved, 9 UGS flows reserved/active
           approximate max available concurrent = 22144 bits (2768 bytes)
```

—continued—

Procedure 10-8 (continued)
Determining upstream bandwidth usage

Calculating supported UGS service flows

1 Calculate the upstream capacity using the following command:

[] Console> **upstream-bandwidth calculate <size> <interval> <us>** ↵

where	is the…
<size>	number of data bytes in a UGS packet
<interval>	grant interval, or packetization rate, in milliseconds
<us>	upstream receiver on which to make the calculation

The CMTS displays current usage and calculations for the specified upstream.

```
20 Theoretical maximum UGS flows (no Initial Maintenance)
17 Theoretical maximum UGS flows (with Initial Maintenance)

9 UGS flows currently active
8 Actual UGS flows can be added based on current usage
```

Note: Typical data sizes are 262 bytes for a 20ms grant interval, or 131 bytes for a 10ms grant interval.

—end—

Managing configuration files and software

Use the procedures in this chapter to back up, restore, and reload configuration files to the CMTS or cable modems.

In this chapter

This chapter contains the procedures listed in Table 11-1.

Table 11-1
Procedures in this chapter

Procedure 11-1
Importing and exporting configuration files

The CLI has three console-level commands which are used for exporting and importing configuration files:

- **putcfg**
- **putmng**
- **getmng**

You can use these commands to save the current CMTS configuration for restoration at a later date, or to display parameters that have been changed from factory defaults.

> *Note:* Due to potential security issues, the **putmng** and **getmng** commands do not output or input user accounts or passwords.

About the putcfg command

The **putcfg** command displays or transfers a configuration file from the CMTS to a specified location. This configuration file contains all writable MIB variables that differ from the CMTS Flash default values.

The output format is for .MD5 configuration records, with each line containing an OID (object ID), a type, a value, and a comment field. The comment field contains the "mib-name.instance" text, so the file can be used with the SNMP set command to configure a device.

> *Note 1:* The MIB variable "lcFlashPersistenceSwitch.0" must be set to **192** (hexadecimal **c0**) to correctly download certain MIB filter items.
>
> *Note 2:* If the IGMP Multicasting feature, with static cache entries, is disabled, errors may occur in the file created by the **putcfg** command.

—continued—

Procedure 11-1 (continued)
Importing and exporting configuration files

About the putmng command

The **putmng** command displays or transfers a configuration file from the CMTS to a specified location. This configuration file contains all writable MIB variables that differ from the CMTS Flash default values, except for user account information.

The output format is in the “manage” format (user-readable text). Each output line is either a level identifier (preceded by a “$”), or a parameter followed by a value.

About the getmng command

The **getmng** command displays or loads a configuration file to the CMTS from a specified location (exported earlier using the **putmng** command). This configuration file contains all settable MIB variables, except for user account information, that differ from the CMTS Flash default values.

The input format is in the “manage” format (see above for details).

Requirements and considerations

The following items are required to perform the tasks in this procedure:

- You must be logged into the CMTS CLI with administrative privileges.
- Make sure the CMTS is displaying the Console> prompt before starting.

Keep the following considerations in mind:

- On UNIX systems, file names are case-sensitive.
- On some UNIX systems, you must set write permissions for the TFTP directory.
- On some systems, you may have to create the destination file before using the **putcfg** or **putmng** commands.
- Make sure the system that you use to transfer files is running a TFTP server.

—continued—

Procedure 11-1 (continued)
Importing and exporting configuration files

Action

Perform the following tasks in this procedure in any order.

Task	Page
Exporting configuration files with the putcfg command	11-4
Exporting configuration files with the putmng command	11-4
Importing configuration files with the getmng command	11-5

Exporting configuration files with the putcfg command

1 Display the current CMTS configuration information, which is stored in flash, using the following command:

[] Console> **putcfg display** ↵

The CMTS displays the configuration file data on the console or remote terminal screen.

2 Upload the current configuration to a TFTP server using the following command:

[] Console> **putcfg <tftpip> <filename>**

where	is the…
<tftpip>	IP address or FQDN of the TFTP server
<filename>	name of the file on the TFTP server that is to contain the configuration data

Exporting configuration files with the putmng command

1 Display the current CMTS configuration information, which is stored in flash, using the following command:

[] Console> **putmng display** ↵

The CMTS displays the configuration file data on the console or remote terminal

2 Upload the current configuration to a TFTP server using the following command:

[] Console> **putmng <tftpip> <filename>**

where	is the…
<tftpip>	IP address or FQDN of the TFTP server
<filename>	name of the file on the TFTP server that is to contain the configuration data

—continued—

Procedure 11-1 (continued)
Importing and exporting configuration files

Importing configuration files with the getmng command

1 Load a configuration file from a TFTP server using the following command:

[] Console> **getmng <tftpip> <filename>**

where	is the…
<tftpip>	IP address or FQDN of the TFTP server
<filename>	name of the file on the TFTP server that contains the configuration data to load into the CMTS

Note: The **getmng** command can temporarily halt CLI operation for up to five minutes if the configuration file is extremely large.

—end—

Procedure 11-2
Upgrading CMTS software using the CLI

The CMTS unit is shipped with system software pre-loaded into non-volatile RAM (NVRAM) in two separate copies ("Image A" and "Image B"). The first image is used to operate the system, while the second is used to retain the default factory settings. The current software version in the system can be viewed through the front panel display, or checked from the CLI or network manager program.

The software image can be upgraded from a TFTP server. The following procedure details the commands required to upgrade the CMTS from the CLI. If you prefer to use a network manager, see "Upgrading CMTS software using an SNMP manager" on page 11-9.

Requirements

You should have the following items and equipment for this upgrade procedure:

- The new CMTS software, either on distribution media or downloaded from the ARRIS FTP site or other location). The software must be placed on a TFTP server so that the CMTS can download it.
- The CMTS should be in an operational network.
- Read the software upgrade "readme" file for the latest released software upgrade information.

Recommendations

These recommendations are not necessary to successfully upgrade the CMTS, but should be followed under normal conditions.

- If you are performing this procedure from the CLI, log into the CMTS through the console port.
- Traffic should be minimal on the network during the upgrade process. After the upgrade has started, the process takes approximately one minute per cable modem or cable modem termination system.

—continued—

Procedure 11-2 (continued)
Upgrading CMTS software using the CLI

Action

Follow these steps to upgrade the CMTS.

> **CAUTION**
> **Network disruption**
> Upgrading the CMTS forces a reset that momentarily disrupts the network. Perform this procedure during a low traffic period.

1 If you downloaded the CMTS upgrade package, make sure the entire distribution is available. The README file contains a list of files included in the distribution.

2 Copy the CMTS upgrade software to the TFTP server’s outbound directory.

Note: This should be the same directory used for the cable modem configuration and .MD5 files.

3 Make sure the CMTS and TFTP server can communicate.

Note: Pinging the CMTS verifies Ethernet connectivity, but does not verify TFTP server connectivity.

4 Log into the CMTS CLI.

The Console> prompt appears.

5 Type the following commands to prepare for the upgrade:

Console> **manage** ↵

[] box# **admin** ↵

[] admin# **info** ↵

The CMTS displays a list of parameters and their current values. For example:

```
Parameter                 Value
provisioning-control      use-both-dhcp-and-tftp
sw-server-ip-addr         0.0.0.0
sw-filename               “CMTS_ALBUM_3.2.0”
config-tftp-ip-addr       0.0.0.0
config-tftp-filename      ""
sw-admin-status           ignore-provisioning-upgrade
dp-statistics-interval    10 seconds
[] admin#
```

—continued—

Procedure 11-2 (continued)

Upgrading CMTS software using the CLI

6 Type the following commands to set the parameters for upgrading.

Note: You can skip any parameters that are already set properly, according to the output of the **info** command in step 5.

```
[] admin# provisioning-control use-both-dhcp-and-tftp ↵
[] admin# sw-server-ip-addr <albumaddr> ↵
[] admin# sw-filename "<name>" ↵
[] admin# config-tftp-ip-addr <tftpaddr> ↵
[] admin# config-tftp-ip-filename "" ↵
[] admin# dp-statistics-interval 10 ↵
```

where	is the…
<albumaddr>	IP address of the server containing the CMTS software load
<name>	name of the upgrade file in the TFTP server public directory (not including the path name); for example, CMTS_ALBUM_4.3.1
<tftpaddr>	IP address of the TFTP server

7 Command the CMTS 1500 to load the upgrade file by entering the following at the admin# prompt:

```
[] admin# sw-admin-status upgrade-from-mgt ↵
```

The upgrade process closes the CLI login session.

When the upgrade completes, the CMTS 1500 sends a new login prompt.

—end—

Procedure 11-3
Upgrading CMTS software using an SNMP manager

The CMTS unit is shipped with system software pre-loaded into non-volatile RAM (NVRAM) in two separate copies (“Image A” and “Image B”). The first image is used to operate the system, while the second is used to retain the default factory settings. The current software version in the system can be viewed through the front panel display, or checked from the CLI or network manager program.

The software image can be upgraded from a TFTP server. This procedure details the steps required to upgrade the CMTS from a network manager. If you prefer to use the CMTS CLI, see “Upgrading CMTS software using the CLI” on page 11-6.

Requirements

You should have the following items and equipment for this upgrade procedure:

- The new CMTS software, either on distribution media or downloaded from the ARRIS FTP site or other location). This software needs to be installed where the CMTS can download it from a TFTP server.
- The CMTS should be in an operational network.
- Read the software upgrade “readme” file for the latest released software upgrade information.

Recommendations

These recommendations are not necessary to successfully upgrade the CMTS, but should be followed under normal conditions.

- Traffic should be minimal on the network during the upgrade process. After the upgrade has started, the process takes approximately one minute per cable modem or cable modem termination system.

—continued—

Procedure 11-3 (continued)
Upgrading CMTS software using an SNMP manager

Action

Follow these steps to upgrade the CMTS.

> **CAUTION**
> **Network disruption**
> Upgrading the CMTS forces a reset that momentarily disrupts the network. Perform this procedure during a low traffic period.

1 If you downloaded the CMTS upgrade package, make sure the entire distribution is available. The README file contains a list of files included in the distribution.

2 Copy the CMTS upgrade software to the TFTP server's outbound directory.

Note: This should be the same directory used for the cable modem configuration and .MD5 files.

3 Make sure the CMTS and TFTP server can communicate.

Note: Pinging the CMTS verifies Ethernet connectivity, but does not verify TFTP server connectivity.

4 Log on to the network manager, if necessary, and select the CMTS to upgrade.

—continued—

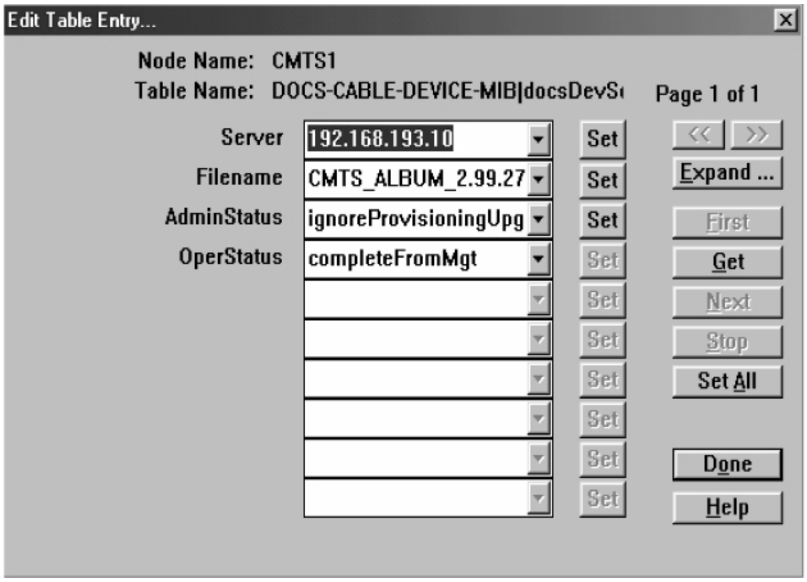
Procedure 11-3 (continued)
Upgrading CMTS software using an SNMP manager

5 Select the **docsDevSoftware** MIB table. See the documentation for your network manager for information about selecting MIB objects and tables.

A table, similar to Figure 11-1, appears.

Figure 11-1
Example docsDevSoftware table

CS-10730



Edit Table Entry...

Node Name: CMTS1
Table Name: DOCS-CABLE-DEVICE-MIB|docsDevS

Page 1 of 1

Server: 192.168.193.10
Filename: CMTS_ALBUM_2.99.27
AdminStatus: ignoreProvisioningUpg
OperStatus: completeFromMgt

6 Set the objects in the **docsDevSoftware** table as follows:

MIB Object	Value
docsDevSwServer	the TFTP server’s IP address
docsDevSwFilename	the file name of the upgrade software (do not include the path) Example: CMTS_ALBUM_4.3
docsDevSwAdminStatus	UpgradeFromMgt

*When you set the AdminStatus object to **UpgradeFromMgt**, the SNMP manager commands the CMTS to begin the upgrade.*

—continued—

Procedure 11-3 (continued)
Upgrading CMTS software using an SNMP manager

7 Monitor the docsDevSwOperStatus object.

If the value is…	**Then** …
InProgress	the CMTS is upgrading its software.
CompleteFromMgt	the upgrade completed successfully.
Failed	the upgrade failed. Look for connectivity problems or a corrupted upgrade file.

After the upgrade completes, the CMTS reboots.

8 Poll the sysObjectID object for the CMTS.

The value should be similar to ***lancityMcnsProdID CMTS.4.3.1.0***

If the CMTS fails to respond after the upgrade procedure, remove the unit from the cable system before performing any troubleshooting procedures.

9 Upgrade and compile the latest SNMP MIBs that are part of this release.

10 Verify that the active modems successfully register with the CMTS. You can verify this by checking the docsIfCmStatusValue MIB object for the modems in question—the status should be "**registrationComplete**(11)" or "**operational**(12)."

—end—

Managing the Enhanced Forwarding Database

This chapter describes the following software features and the procedures for configuration via the CLI using the **manage** and **set** (SNMP) commands:

- Enhanced Forwarding Database (FDB)
- Cable Modem Aging Changes
- Provisionable Age-out Interval for CMs that have Ranged but not Registered
- Provisionable Age-out Interval for CMs that have De-registered
- Mode B Forwarding
- Mechanism for Preventing Address Request Protocol (ARP) Spoofing
- Funnel Mode

Forwarding Database Feature Enhancement

The Forwarding Database (FDB) has been enhanced to increase its performance in flat networks. Since subscribers are implementing home networking, with multiple CPEs behind the CMTS, the CMTS FDB has been enhanced to learn more devices, with entries reserved specifically for cable modems (CMs).

A special command, (**purge-fdb**), has been added to the CLI to prevent FDB overflows. For example, if you have a flat network in which a device is not being learned, you may want to purge the FDB to clean out the old devices no longer in service, thereby allowing new devices to be learned.

The **purge-fdb** command purges all Ethernet and Customer Provided Equipment (CPE) entries from the FDB. The **purge-fdb** command requires read-write access and is not available via SNMP.

CAUTION
Service-affecting
Purging the FDB will disrupt all data traffic through the CMTS.

The following warning message is displayed prior to executing the purge-fdb command:

```
This will cause all the non-CM dynamic entries in the FDB to be deleted,
and influence traffic through the CMTS by stopping all packet processing
until the purge is complete. Do you really want to perform the purge (y/n)?
```

CM MAC addresses are learned by the FDB from Ranging Request MAC Management messages and IP packets, ensuring that CM devices are identified correctly and as soon as possible. If a CM device attempts to register when the maximum number of CM devices are already registered, the CMTS refuses to register the device and reports this event.

If a CPE device is learned after the maximum number of non-CM devices has been added to the FDB, the CPE device will not be added to the FDB. Furthermore, if there is no FDB entry for the CPE, then no association is made between the device and the CM device. Therefore, baseline privacy is not incorporated in the communication from the CMTS down to the CPE device, and a lookup cannot be performed to find the CM given the CPE information or vice versa.

Cable Modem Aging Changes

The aging characteristics of CM entries in the FDB have been modified to improve the performance of the FDB. In order to free up FDB CM entries as quickly as possible, the following new features have been added:

- Provisionable age-out interval for CMs that have ranged but not registered.
- Provisionable age-out interval for CMs that have de-registered.

Provisionable Age-out Interval for CMs that have Ranged but not Registered

A user is able to provision the duration of the aging timer for failed CM registrations from 30 seconds to 432,000 seconds (5) days via the following. The default is 120 seconds (2 minutes).

- CLI using the **manage** command
- CLI using the **set** (SNMP) command

CLI Using Manage Command

To provision the duration of the aging timer for failed CM registrations in the CLI using the **manage** command, enter the following:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# ranged-only-cm-aging-time <value> ↵
```

where value equals (30 seconds to 432,000 seconds).

CLI Using Set (SNMP) Command

A new MIB object, lcRngNotRegCmAgeTime, has been created in the LanCity MIB.

To provision the duration of the aging timer for failed CM registrations in the CLI using the set command, enter the following:

```
Console> set lcRngNotRegCmAgetime.0 <value> ↵
```

where value equals (30 seconds to 432,000 seconds)

Provisionable Age-out Interval for CMs that have De-registered

A user is able to provision the duration of the aging timer for CMs that have de-registered from 30 seconds to 432,000 seconds (5 days) via the following. The default is 172,800 seconds (48 hours).

- CLI using the manage command
- CLI using the set (SNMP) command

CLI Using Manage Command

To provision the duration of the aging timer for CMs that have de-registered in the CLI using the manage command, enter the following:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# deregistered-cm-aging-time <value> ↵
```

where value equals (30 seconds to 432,000 seconds)

CLI Using Set (SNMP) Command

A new MIB object, **lcDeregCmAgeTime**, has been created in the LanCity MIB.

To provision the duration of the aging timer for CMs that have de-registered in the CLI using the set command, enter the following:

```
Console> set lcDeregCmageTime.0 <value> ↵
```

where value equals (30 seconds to 432,000 seconds)

Mode B Forwarding

Mode B Forwarding allows the CMTS software to incorporate Layer 3 (IP Layer) Switching and cable modem (CM) entries to be aged-out (eliminated after a pre-defined time limit) of the FDB based on the Dynamic Host Configuration Protocol (DHCP) lease time.

The default configuration for the CMTS is Mode A, where the MIB object, **lcForwardingMode** is set to **none**. Mode A is a standard learning bridge in which packets are forwarded based on MAC addresses. In Mode A there is no aging of CM entries from the FDB as long as they are registered.

One of the benefits of Mode B Forwarding is that it allows the CMTS to act as an ARP proxy by responding to all known IP/MAC table entries on the cable side, thereby reducing broadcast traffic. Proxy ARP is automatically enabled with Mode B where **lcForwardingMode** is set to **dhcp-arp**. Another benefit of Mode B Forwarding is that it provides filtering and security in order to track IP addresses.

Mechanism for Preventing Address Request Protocol (ARP) Spoofing

The mechanism for preventing ARP spoofing prevents anyone outside from making changes to the IP address of a CPE device from its DHCP-assigned value and disseminating it throughout the network. When the mechanism for ARP spoofing is enabled, the feature prevents IP address stealing on the HFC side of the CMTS network.

Funnel Mode

Funnel Mode allows you the capability to monitor traffic flow in order to comply with law enforcement agency requests and other traffic monitoring applications.

In a CMTS default configuration, funnel mode is disabled and packets are processed in normal mode. Any packets that are sent from a CM destined for another CM on the same CMTS are intercepted by the CMTS and sent back down the cable and not out on the Ethernet.

When funnel mode is enabled, packets received on the upstream RF interface are destined for an RF-side device, are forwarded to the Ethernet port where it is the responsibility of the switch or other device (for example, a Shasta box) to send the packet to its destination.

In this chapter

This chapter contains the procedures listed in Table 12-1.

Table 12-1
Procedures in this chapter

Procedure	Title	Page
12-1	Configuring Mode B Forwarding	12-6
12-2	Configuring ARP Spoofing	12-7
12-3	Configuring Funnel Mode	12-8

Requirements

Be sure the CMTS is displaying the Console> prompt before starting a procedure.

Procedure 12-1
Configuring Mode B Forwarding

The following procedures explain how to enable/disable Mode B Forwarding in the:

- CLI using the **manage** command
- CLI using the **set** (SNMP) command

Configuring Mode B Forwarding via the CLI

1 Enter the following commands to enable Mode B Forwarding via the CLI:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# info ↵
[] forwarder# forwarding-mode ? ↵
[] forwarder# forwarding-mode dhcp-arp ↵
```

2 Type **exit** to return to the Console> prompt.

3 Enter the following commands to disable Mode B Forwarding via the CLI:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# info ↵
[] forwarder# forwarding-mode ? ↵
[] forwarder# forwarding-mode none ↵
```

4 Type **exit** to return to the Console> prompt.

Configuring Mode B Forwarding via the set (SNMP) command

The MIB object, **lcForwardingMode**, is contained in the lccmtsDPConfiguration Info Table. The options are:

- **none** (1)—The CMTS uses transparent learning (Mode A).
- **dhcp-arp** (2)—The CMTS uses Layer 3 switching based on DHCP and ARP.

1 Enter the following command to enable Mode B Forwarding in the CLI using the **set** command:

```
Console> set lcForwardingMode.0 2 ↵
```

2 Enter the following command to disable Mode B Forwarding in the CLI using the **set** command:

```
Console> set lcForwardingMode.0 1 ↵
```

—end—

Procedure 12-2
Configuring ARP Spoofing

The following procedures explain how to enable/disable ARP Spoofing in the:

- CLI using the **manage** command
- CLI using the **set** (SNMP) command

Configuring ARP Spoofing via the CLI:

1 Enter the following commands to enable ARP Spoofing via the CLI:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# arp-spoofing-protection ? ↵
[] forwarder# arp-spoofing-protection enable ↵
```

2 Type **exit** to return to the Console> prompt. ↵

3 Enter the following commands to disable ARP Spoofing via the CLI:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# arp-spoofing-protection ? ↵
[] forwarder# arp-spoofing-protection disable ↵
```

4 Type **exit** to return to the Console> prompt. ↵

Configuring ARP Spoofing via the set (SNMP) command

The MIB object, **lcArpSpoofingProtection**, is contained in the lccmtsDPConfiguration Info Table. The options are:

- **enable** (1)—The CMTS does not process ARP packets received from the cable interface, if the sender's address information does not match the internal CMTS FDB.
- **disable** (2)—The CMTS forwards the ARP packet and updates the ARP cache with the ARP IP address.

1 Enter the following command to enable ARP spoofing in the CLI using the **set** command:

```
Console> set lcArpSpoofingProtection.0 1 ↵
```

2 Enter the following command to disable ARP spoofing in the CLI using the **set** command:

```
Console> set lcArpSpoofingProtection.0 2 ↵
```

—end—

Procedure 12-3
Configuring Funnel Mode

The following procedures explain how to enable/disable Funnel Mode in the:

- CLI using the **manage** command
- CLI using the **set** (SNMP) command

Configuring Funnel Mode via the CLI

1 Enter the following commands to enable Funnel Mode via the CLI:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# info ↵
[] forwarder# funnel-mode enable ↵
```

2 Type **exit** to return to the Console> prompt.

3 Enter the following commands to disable Funnel Mode via the CLI:

```
Console> manage ↵
[] box# forwarder ↵
[] forwarder# info ↵
[] forwarder# funnel-mode disable ↵
```

4 Type **exit** to return to the Console> prompt.

Configuring Funnel Mode via the set (SNMP) command

The MIB object, **lcArpFunnelMode**, is contained in the lccmtsDPConfiguration Info Table. The options are:

- **enable** (1)—Packets received on the upstream RF interface destined for an RF-side device, are forwarded to an Ethernet port.
- **disable** (2)—Packets are processed in the normal (default) mode.

1 Enter the following command to enable Funnel Mode in the CLI using the **set** command:

```
Console> set lcFunnelMode.0 1 ↵
```

2 Enter the following command to disable Funnel Mode in the CLI using the **set** command:

```
Console> set lcFunnelMode.0 2 ↵
```

—end—

Glossary

Alarms

Messages specific to the Cornerstone products, to report events not available in the DOCSIS specifications (such as loss of communications).

Authentication protocol

A mechanism in SNMPv3, used to guarantee authorization to individual users.

Best Effort

A Service Flow type suited for typical interactive data transfers (web browsing, low-priority file transfers, and the like). It does not provide bandwidth guarantees. A Best Effort flow requests “grants” as needed, and may not actually receive the grants depending on traffic and priority levels.

C3M

Customer Controlled Cable Modem, a device consisting of (usually) a card that plugs into the subscriber’s computer, and cable modem software running on the subscriber’s computer.

Carrier path

The physical upstream route from a group of cable modems to the CMTS.

Certificate

Similar to a key, but used to authenticate cable modems (and similar devices) for use on the network.

Channel group

A group of upstreams with separate physical paths, but share a common upstream frequency and modulation characteristics.

Class of Service

A DOCSIS 1.0 service that allows you to specify maximum (and minimum) bandwidth values and priorities. See “Class of Service (CoS)” on page 9-3 for details.

Classifier

Rules used to classify packets into a Service Flow. The device compares incoming packets to an ordered list of rules at several protocol levels. Each rule is a row in the docsQosPktClassTable.

A matching rule provides a Service Flow ID (SFID) to which the packet is classified. All rules need to match for a packet to match a classifier. Packets that do not match any classifiers are assigned to the default (or primary) Service Flow.

Client

A system that a remote operator uses to connect to the CMTS using SSH.

Committed Information Rate (CIR)

A service type (not a strict Service Flow) provided by specifying a minimum reserved traffic rate with a Service Flow type such as Best Effort or nrtPS.

Community string

A character string, similar to a password, that allows access to one or more network objects by a network management system.

Concatenation

Concatenation is an upstream capability of the cable modem, which allows the cable modem to combine multiple packets into one large packet for upstream transmission.

This capability must be supported in both the CMTS and the cable modem.

CPE (or Subscriber host)

Customer Premises Equipment—any subscriber device intending to use the cable data network for communications.

docsQosCmtsMacToSrvFlowTable

Lists the Service Flow IDs associated with a particular cable modem MAC address. This allows for indexing into other docsQos tables that are indexed by docsQosServiceFlowId and ifIndex.

docsQosDynamicServiceStatsTable

Describes statistics associated with the Dynamic Service Flows in a managed device.

docsQosParamSetTable

Describes set of DOCSIS QoS parameters defined in a managed device. Each entry defines one QoS Parameter Set.

docsQosPktClassTable

Describes the packet classification configured in the cable modem or CMTS (“Classifier” table).

docsQosServiceFlowLogTable

Contains a log of the disconnected Service Flows in a managed device.

docsQosServiceFlowStatsTable

Describes statistics associated with the Service Flows in a managed device.

docsQosServiceFlowTable

Describes a set of DOCSIS QoS Service Flows in a managed device.

docsQosUpstreamStatsTable (CMTS only)

Describes statistics associated with upstream Service Flows.

Downstream

The RF (and fiber) path from the CMTS to a cable modem.

Events

Error and information messages; you can configure the CMTS to report events using SNMP traps or syslog services (or both).

Fragmentation

Fragmentation is an upstream capability of the cable modem, that allows the cable modem to split large packets into smaller packets for more efficient use of upstream bandwidth.

This capability must be supported in both the CMTS and the cable modem per Service Flow. The cable modem must be provisioned for fragmentation through the .MD5 configuration file.

Host

The CMTS.

Host key

Actually a pair of keys, one public (sent to clients) and one private (not distributed) that the CMTS uses to decrypt incoming traffic.

Ingress Noise

Undesired RF energy entering the cable plant, especially the upstream, from an outside source.

Ingress avoidance action

How the CMTS reacts to ingress beyond a certain level; this can involve shifting the upstream frequency or using lower data rates that can better tolerate the ingress.

Jitter

The maximum difference in the interval between packets. Certain services, such as telephony, require minimal jitter.

Key

A block of bits, used to encrypt or decrypt data on an SSH link. Keys can be from 512 to 2048 bits long, in increments of 128 bits. Longer keys are harder to decrypt by someone attempting to compromise your security, but can seriously impact CMTS performance.

Metric

A method for determining whether an ingress avoidance action is necessary or imminent.

Modulation profile

A defined modulation type and forward error correction (a modulation profile can specify other parameters but these are not important for ingress avoidance).

Non-real-time Polling Service (nrtPS)

A Service Flow type suitable for high-bandwidth, jitter-tolerant services such as file transfer.

Notifications

Messages used by SNMPv2/v3 manager programs in a manner similar to error messages and warnings. Notifications are actually part of the MIB tree, and have OID values. Notifications are analogous to SNMPv1 traps.

Object identifier (OID)

The internal representation of a MIB object. This is a string of decimal numbers separated by periods, such as **1.3.6.1.2.1.1.1.4.1**.

Primary Service Flow

The default path for maintenance traffic and any packets not classified onto any other Service Flow. All cable modems have a Primary Upstream Service Flow and a Primary Downstream Service Flow.

Privacy protocol

A mechanism in SNMPv3, used to encrypt data traffic between a network management system and the managed device.

Real-time Polling Service (rtPS)

A Service Flow type suited for services using variable packet sizes at periodic intervals, such as MPEG video.

Server key or session key

A key that the CMTS creates for each SSH session to encrypt outgoing traffic. This key is regenerated periodically for added security.

Service Flow

A unidirectional flow of packets, used to define and shape traffic. See “Service flows” on page 9-5 for details.

Service ID (SID)

A number, assigned by the CMTS, to each upstream data link; the SID is used for a number of purposes including BPI.

Static provisioning

A method for provisioning QoS in which each cable modem reads its QoS definitions

Transmission profile

Consists of a modulation type and bandwidth; ingress avoidance actions work by changing the current avoidance profile.

Traps

Messages sent when certain events occur, and are generated by SNMPv1 devices. Traps do not have an object id (OID) associated with them, but typically display text messages.

Unsolicited Grant Service (UGS)

A Service Flow type suited for services requiring constant bandwidth, such as telephony. The flow is given grants at regular intervals without needing to request them.

Unsolicited Grant Service with Activity Detection (UGS/AD)

A Service Flow type suited for services requiring constant bandwidth, but with intervals of inactivity such as telephony. The flow is given grants at regular intervals unless the upstream is inactive. This allows bandwidth to be released for Best Effort and similar traffic when not required for the higher-priority services.

Upstream

The RF (and fiber) path from a cable modem to the CMTS.

Index

Cornerstone

CMTS 1500 User Guide

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Arris Interactive.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license. A summary of the Software License is included in this document.

Document number: ARSVD00729
Release 4.3   Standard 1.0
September 2003
Printed in the USA

ARRIS™