ARRIS™

Cadant C3 CMTS

Installation, Operation, and Maintenance Guide

Release 3.0   Standard 2.0   March 2004

ARSVD00814

Cadant C3 CMTS

Installation, Operation, and Maintenance Guide

Document number: ARSVD00814
Document release: Release 3.0 Standard 2.0
Date: March 2004

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.

Publication history

March 2004 Release 3.0 Standard 2.0 version of this document for version 3.0.

August 2003 Release 2.0 Standard 1.0 version of this document.

Contents

Bridge operation 3-1

Providing Multiple ISP Access 4-1

IP Routing 5-1

Command Line Interface Reference 6-1

Configuring Security 8-1

Service Procedures 9-1

Configuration Forms D-1

Glossary E-1

About this Manual

This document provides necessary procedures to install, operate, and troubleshoot the ARRIS Cadant C3 CMTS in a DOCSIS®-compatible environment.

Scope

This document is intended for cable operators and system administrators who configure and operate the CMTS. It is assumed the reader is familiar with day-to-day operation and maintenance functions in networks that rely on TCP/IP protocols and hybrid fiber/coax (HFC) cable networks.

In this Document

This manual provides the following content:

- Chapter 1, “Getting Started,” provides a brief overview of the Cadant C3 CMTS and its components.
- Chapter 2, “CMTS Installation,” describes how to unpack and install the CMTS including how to bring up the CMTS from an “out of box” condition to full operation.
- Chapter 3, “Bridge operation,” describes basic bridge operation of the CMTS and issues in upgrading to L3 capable code to restore DHCP operation.
- Chapter 4, “Providing Multiple ISP Access,” describes the supported 802.1Q VLAN capabilities.
- Chapter 5, “Layer 3 operation,” describes how to configure the C3 CMTS as a layer 3 router.
- Chapter 6, “Command Line Interface Reference,” describes the command line interface for managing and configuring the CMTS.

- Chapter 7, “Managing Cable Modems,” describes common procedures for operating and troubleshooting DOCSIS systems.
- Chapter 8, “Configuring Security,” describes methods that can be used to improve security of management and user traffic.
- Chapter 9, “Service Procedures,” describes basic service procedures.
- Appendix A, “Specifications,” lists physical, electrical, and networking specifications.
- Appendix B, “CMTS Configuration Examples,” provides a configuration for a bench top trial. Includes both RF and CLI configuration.
- Appendix C, “Factory Defaults,” contains default configuration information.
- Appendix D, “Configuration Forms,” provides a form listing essential configuration parameters.
- Appendix E, “Glossary,” provides a glossary of terms used in this manual.

Conventions Used in This Manual

Various fonts and symbols are used in this manual to differentiate text that is displayed by an interface and text that is selected or input by the user:

Highlight	Use	Examples
bold	Keyword: Text to be typed literally at a CLI prompt.	Type **exit** at the prompt.
italics	In commands, indicates a parameter to be replaced with an actual value.	**ping {ipaddr}**
bracketed	A parameter in a CLI command. A parameter enclosed in [square] brackets is optional; a parameter enclosed in {curly} brackets is mandatory.	**ping {*ipaddr*}** **terminal [no] monitor**
monospaced	Display text. Shows an interactive session of commands and resulting output.	
ipaddr	IP address: enter an IP address in dotted-quad format	10.1.105.128
macaddr	MAC address: enter a MAC address as three 4-digit hexadecimal numbers, separated by periods.	00a0.731e.3f84
⚠	Caution: Indicates an action that may disrupt service if not performed properly.	
⚠	Danger: Indicates an action that may cause equipment damage, physical injury, or death if not performed properly.	
↗	Procedure: Indicates the beginning of one or more related tasks.	

For More Information

For more detailed information about DOCSIS, refer to the following technical specifications, available online at www.cablelabs.com.

- Radio Frequency Interface (RFI) Specification—defines how data is passed over the cable
- Operations Support System Interface (OSSI) Specification—defines how DOCSIS components can be managed by the cable operator
- Baseline Privacy Interface (BPI) Specification—defines how data is encrypted while traveling on the cable to keep it private
- Computer to Modem Communications Interface (CMCI) Specification—defines how PCs can communicate to cable modems

FCC Statement

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

There is no guarantee that interference will not occur in a particular installation. If this device does cause harmful interference to radio or television reception, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the computer and receiver.
- Connect the computer into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Any changes or modifications not expressly approved by the grantee of this device could void the user’s authority to operate the equipment.

Safety

Normal lightning and surge protection measures are assumed to have been followed in the RF plant that the ARRIS Cadant C3 CMTS RF input and output is connected to.

If AC supply is used to power the ARRIS Cadant C3 CMTS, suitable surge and lightning protection measures should be taken with this supply.

The equipment rack the ARRIS Cadant C3 CMTS is mounted in should have a separate safety ground connection. This ground should be wired in accordance with National Electric Code (NEC) requirements for domestic applications and paragraph 2.6 of EN60950/IE950 for international applications.

The safety ground wire must be #6 AWG or larger, and it must connect the equipment rack directly to the single-point ground in the service panel. The single-point ground can be an isolated ground or the AC equipment ground in the service panel or transformer. Depending on the distances between the cabinets and the location of the service panel, the wiring can be either daisy-chained through the cabinets or run independently from each cabinet to the service panel.

The remaining non-RF and non-AC supply connections of the ARRIS Cadant C3 CMTS should be made by SELV rated circuits.

1 Getting Started

This chapter introduces the ARRIS Cadant C3 Cable Modem Terminating System (CMTS) and provides background information about the Data-Over-Cable Service Interface Specification (DOCSIS) standards with which the product complies.

About the C3 CMTS

ARRIS has designed the C3 specifically for DOCSIS and EuroDOCSIS specifications.

From its inception, it has been designed to take advantage of already defined Advanced Physical Layer features as well as new noise suppression technologies to deliver the most efficient utilization of the upstream spectrum. The hardware platform itself has been designed to scale to the most demanding needs of the operator from a packet classification and features perspective. The processing power of the system is capable of accommodating the emerging needs of cable operators worldwide.

With dual RISC processors in its architecture, the C3 supplies the processing power needed to support high volumes of traffic, with excellent latency control. The CMTS has scalable transmit and receive capacity, which can be configured to support one channel downstream and up to six channels upstream. It supports multiple network protocols, and multiple architectures such as PPPoE and NetBEUI, making it easy to add to existing router- or switch-based cable networks. Easy-to-use system management tools include an industry-standard command-line interface.

DOCSIS Compliance

The C3 is DOCSIS 1.1 and EuroDOCSIS 1.1 qualified. The C3 does not support SCDMA and thus is unable to be qualified for DOCSIS 2.0 at this time.

The CMTS works on any cable system with any modems which comply with the DOCSIS specification.

Fast Start

The basics of commissioning the Cadant C3 CMTS are covered in Chapter 2 and a complete example of a bench top installation is also provided in Appendix B.

Introducing the ARRIS Cadant C3 CMTS

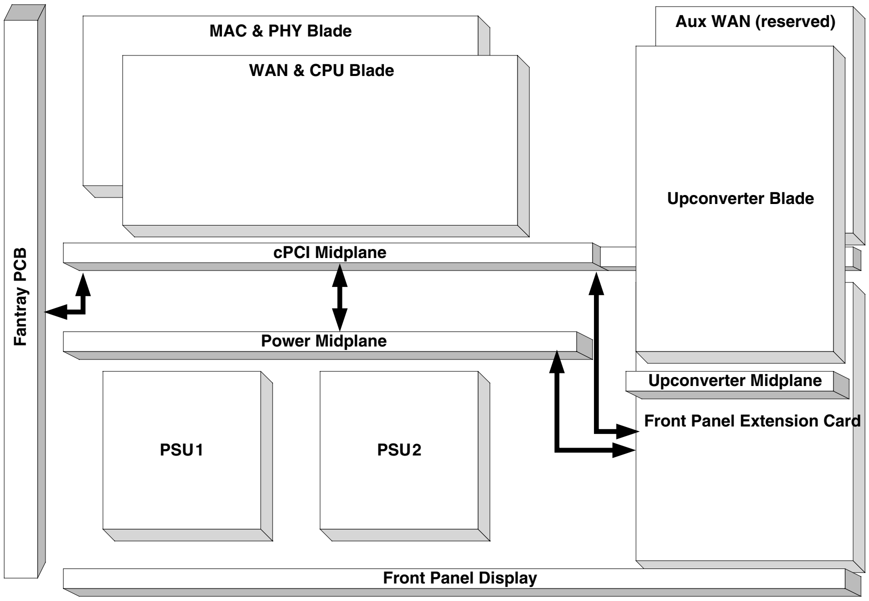
The C3 is a flexible, powerful, and easy-to-use Cable Modem Termination System (CMTS). It is qualified as fully compliant with the DOCSIS 1.1 standards, which includes specifications for features such as security enhancements, telephony, QoS, and tiered services.

The C3 has dual 10/100/1000 Mbps Ethernet interfaces and supports a 64 or 256 Quadrature Amplitude Modulation (QAM) cable TV downstream channel, and up to six variable-rate Quadrature Phase Shift Keying (QPSK) or 8, 16, 32, or 64 QAM upstream channels. Easy-to-use system management tools include an industry-standard command-line interface.

Features	Benefits
Advanced TDMA support: 8QAM, 32QAM, and 64QAM 200 KHz to 6.4 MHz channel width	Designed from the ground up to support advanced symmetrical data rate applications based on the DOCSIS 1.0, 1.1, and 2.0 specifications while maintaining compatibility with existing modems. Delivers superior performance in real-world cable plants through advanced noise cancellation technology
Compact size	Full DOCSIS 1.1 with ATDMA support in a one-rack unit high system
Operator selectable Layer 2 and Layer 3 forwarding	Allows operators to choose the routing method most appropriate to their needs
ACL support	Up to 30 ACLs with 20 entries per ACL may be applied to any interface
Full upstream support 5 to 65 MHz	Allows better utilization of upstream frequency space for DOCSIS in plants outside of North America
DOCSIS and Euro-DOCSIS support—selectable in software	Provides flexibility for operators by supporting either protocol on the same unit with no additional hardware to purchase

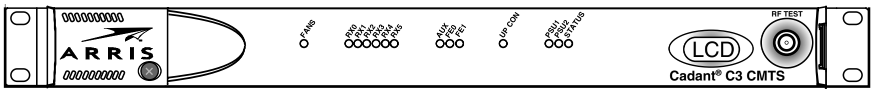
Features	Benefits
Efficient bandwidth management	User-configurable dynamic upstream channel bandwidth allocation allows the ARRIS Cadant C3 to respond to network conditions in real-time. Load-balancing allows the cable operator to automatically or manually distribute upstream traffic evenly across available channels.
Integrated RF up-converter	Complete ready-to-use CMTS in only one rack unit (1.75 in. of space)

The following diagram shows the major components of the Cadant C3 CMTS.



Front panel

The following diagram shows the C3 front panel.



The following table lists and describes the front panel indicators.

Name	Indication	Description
FANS	Green	Normal operation.
	Red	One fan has failed.
	Flashing Red	More than one fan has failed.
RX0 to RX5	Green	Upstream is active.
	Flashing Green	Upstream is in use.
AUX		not used
FE 0	Green	WAN network port is linked.
	Flashing Green	WAN network port is active.
FE 1	Green	MGMT network port is linked.
	Flashing Green	MGMT network port is active.
UP CON	Green	Upconverter is operating properly.
	Off	Upconverter not installed.
PSU 1	Green	Power supply 1 (on the left side behind the front panel) is operating properly.
	Flashing Red	Power supply 1 fault detected.
PSU 2	Green	Power supply 2 (on the right side behind the front panel) is operating properly.
	Flashing Red	Power supply 2 fault detected.
STATUS	Flashing Amber	CMTS is booting.
	Green	Normal operation.
	Flashing Red	CMTS fault detected.
RF test		Downstream output with signal level attenuated by 30 dB

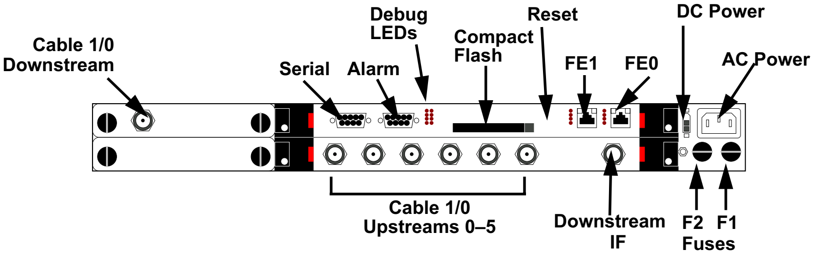
Traffic LED flash rates

The Traffic LED flashes at variable rates to indicate the relative amount of data flowing through the CMTS. The following table interprets the LED flash rate.

Traffic Rate	Flash Rate
>2000 packets per second	50 milliseconds
>1000 packets per second	100 milliseconds
>500 packets per second	150 milliseconds
>300 packets per second	200 milliseconds
>100 packets per second	250 milliseconds
>10 packets per second	300 milliseconds
less than 10 packets per second	500 milliseconds
0 packets per second	not flashing

Rear Panel

The following diagram shows the locations of ports on the rear panel.



The following table describes the ports on the rear panel.

Port	Interface
FE1	10/100/1000Base-T interface
FE0	10/100/1000Base-T interface
AC power	Input receptacle for 90 to 264 volts AC
DC power	Input receptacle for –40 to –60 volt DC
RS232	RS-232 serial port for initial setup (38400/N/8/1)
Alarm	see “Alarm Port” on page 1-6
RX0	Upstream #1 (cable upstream 0)
RX1	Upstream #2 (cable upstream 1)
RX2	Upstream #3 (cable upstream 2)
RX3	Upstream #4 (cable upstream 3)
RX4	Upstream #5 (cable upstream 4)
RX5	Upstream #6 (cable upstream 5)

Port	Interface
Downstream	Downstream output from upconverter
Downstream IF Output	Intermediate frequency (IF) output (43.75 MHz for NA DOCSIS; 36.125 MHz for EuroDOCSIS) which may be routed to an external upconverter.

Note: ARRIS does not support simultaneous use of the Downstream and Downstream IF outputs.

Alarm Port

Reserved for future use.

Major Components of the Cadant C3 CMTS

Redundant Power Supplies

The Cadant C3 CMTS supports simultaneous powering from AC or DC using one or two power supplies. If two power supplies are installed, the load is shared between both. In this configuration, one power supply may fail without impacting system operations. The CMTS has separate connections for AC and DC power.

Up-Converter

The Cadant C3 CMTS incorporates a state-of-the-art up-converter for the downstream signal. The signal may be output in either the DOCSIS (6 MHz wide—Annex B) or EuroDOCSIS (8 MHz wide—Annex A) formats and this format can be configured through software. The integrated up-converter is field-replaceable, and can generate the full DOCSIS/EuroDOCSIS power range across the entire frequency. The up-converter is frequency agile and can be readily tuned either through the command line interface or SNMP.

The CMTS is capable of using various frequency plans, including North American Standard, IRC, HRC, Japanese, European PAL, and European SECAM. For more information on supported channel plans, see Appendix B. The C3 can operate at any frequency (in 62.5 KHz steps) within the band.

Wideband Digital Receiver

The CMTS incorporates a wideband digital receiver for each upstream channel. The digital receiver section allows spectrum analysis as well as advanced digital signal processing to remove noise (including ingress) and deliver the highest possible performance.

Media Access Control (MAC) Chip

The MAC chip implements media access control (MAC) protocol and handles MPEG frames. It also supports Direct Memory Access (DMA) for high data transfer performance.

Ethernet Interfaces

The CMTS has two Ethernet interfaces, each which is capable of operating at 10, 100, or 1000 megabits per second. The ports are capable of both half-duplex and full-duplex operation and automatically negotiate to the appropriate setting. One port may be dedicated to data while the other port may be used for out-of-band management of the C3 and (optionally) cable modems.

Management Schemes

The CMTS management mode determines how traffic is assigned to the Ethernet ports, and may be selected through the C3 configuration. For example:

- C3 management traffic can be restricted to one Ethernet port, and all subscriber traffic restricted to the other Ethernet port.
- Cable modem traffic can be directed to either Ethernet port as required.

CPU

The CMTS is built around dual, state-of-the art, reduced instruction set (RISC) processors. One processor is dedicated to data handling while the other processor performs control functions including SNMP.

Flash Disk

The C3 uses a SanDisk 128MB Compact Flash card to store operating software and configuration files. The disk may be removed without affecting normal operation; however, the C3 disables all configuration-related CLI and SNMP functions until you replace the disk.

ARRIS recommends using SanDisk 128MB or 256MB Compact Flash cards with the C3 CMTS. While other brands of Compact Flash cards may also work, ARRIS cannot guarantee their proper operation in the C3.

2 CMTS Installation

Use this chapter to install the Cadant C3 CMTS.

Planning the Installation

Network Requirements

The CMTS may be connected to your network using one or both Ethernet interfaces. Use the following table to determine the best configuration for your installation.

If you want to…	Then use…
physically separate management traffic from data traffic	both Ethernet interfaces.
separate management traffic from user traffic	both Ethernet interfaces or a single Ethernet interface and VLANs (see Chapter 5).

Regardless of the connection method selected, at least one network connection is required to the CMTS.

Network interaction

How the ARRIS Cadant C3 is to interact with the network is another consideration.

- Simple bridging operation with one cable sub-interface and one fastethernet sub-interface configured within a single bridge-group.
- Simple bridging operation with two fastethernet sub-interfaces (one on each fastethernet port), and one cable sub-interface configured within a single bridge-group. Depending on network configuration this option may require DHCP RELAY to be activated.
- Complex bridging operation with bridge groups linking multiple cable and Fast Ethernet sub-interfaces and optionally using 802.1Q VLANs.

- Layer 3 routing, routing between multiple cable and Fast Ethernet sub-interfaces, optionally using 802.1Q VLANs

Sub-interfaces and their use are explained fully in Chapter 4 as is optional routing operation of the ARRIS Cadant C3.

Power Requirements

To assure high system reliability, the C3 chassis supports two hot-swappable, load-sharing power supply modules. A single supply can provide all the power that a fully loaded system needs with sufficient safety margin.

Each type of power supply has a separate power connector mounted on the rear panel of the C3 chassis. The power connectors are typically plugged into the AC power or DC power distribution unit of the rack or cabinet using the power cords supplied with the C3.

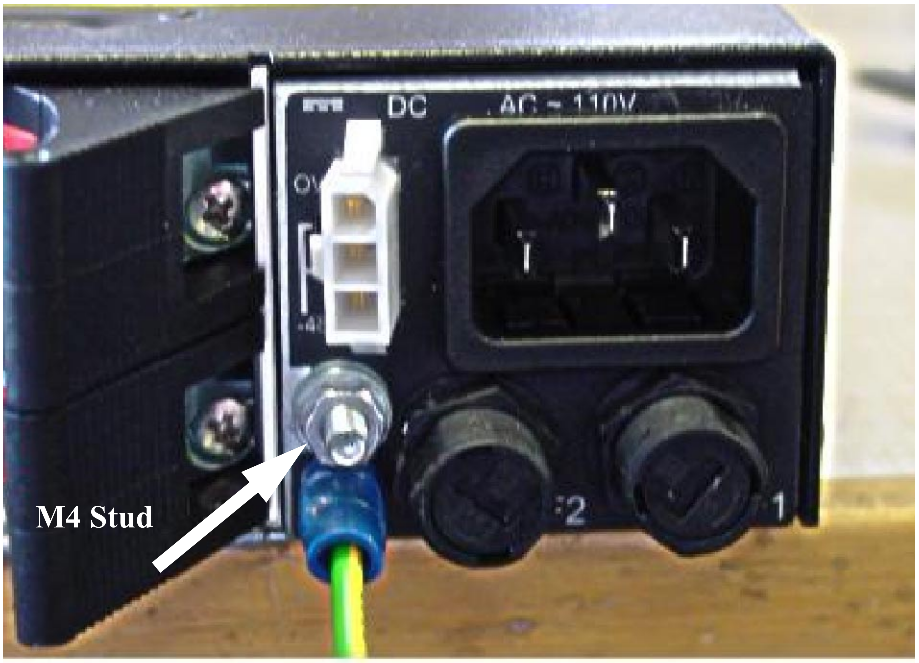
> ***Note:*** Make sure that the power circuits have sufficient capacity to power the C3 before connecting power.

To disconnect power from the C3 for servicing, remove both power leads (AC and DC) from the rear socket. The C3 has no power switch.

Earthing

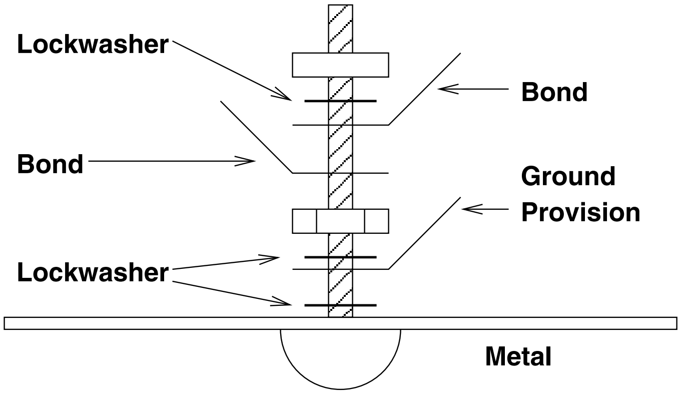
Reliable earthing of rack mounted equipment should be maintained. See “Safety” on page xxiii for common safety considerations. Also consider using power strips instead of direct connections to branch circuits.

When using only DC power, earth the C3 chassis using the supplied M4 stud.



Use an M4 nut and M4 lock washers with the parts stacked as shown in the figure below.

The power supply cord binding conductor may be secured either under the first (bottom) nut or the second (top) nut since replacement of either the power supply cord or the component being handled could occur first.



AC powering

The AC power modules require 100 to 240 volt, 2A, 47 to 63 Hz AC power. The socket-outlet must be properly earthed.

DC powering

The DC power modules requires –40 to –60 V DC, 4A power from a SELV rated source.

The DC power source must have an over current protection device rated at 10 Amp.

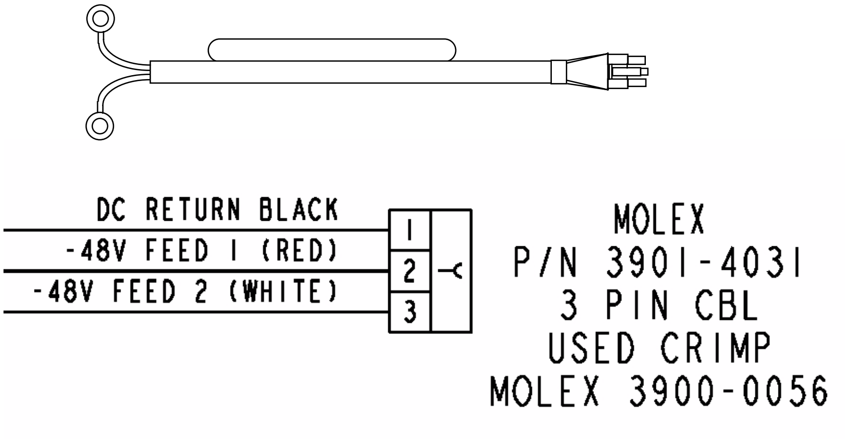
Connect the supplied external DC cable assembly to –48V DC using a Carling Technologies Inc. Part Number LDC1-AL-10-10-10-10-10-10-J power distribution unit as shown following.

The external DC cable assembly must not be modified in the field; route any excess length to avoid snags.

Connect both Feed 1 and Feed 2 to –48V even if only one DC power supply is to be installed. This allows placing a single DC power supply in either of the two possible locations, or placing two DC power supplies in the chassis.

Signal	To	AWG	Color
DC Return	Pin 1	18	Black
–48V Feed 1	Pin 2	18	Red
–48V Feed 2	Pin 3	18	White

The following diagram shows the connector and pin locations.



Cable Requirements

A variety of cables and connectors and the tools to work with them must be obtained to complete the installation. The following table shows the cable and connector types.

Cable	Wire Type	Connector Type
Serial console (included with C3)	9 pin RS-232 serial cable	DB-9M
Ethernet connections	Category 3, 4, 5, or 5E twisted pair cable	RJ-45
CATV	RG-59 or RG-6 (RG-6 recommended)	F

Ethernet Connections

The C3 provides two 10/100/1000BaseT Ethernet ports to allow connection to a terminating router, server, or other networking devices such as a hub, switch, or bridge.

Both Ethernet connectors are standard RJ-45 connectors. For 10BaseT and 100BaseT, unshielded cable may be used. For 1000BaseT, use shielded category 5E wire.

Cable Plant Requirements

The RF cable plant should be designed so that all RF ports connect to SELV circuits (meeting the requirements of SELV as defined in UL60950). You must provide suitable protection between these ports and the CATV outside plant.

Downstream RF cable plant requirements are as follows:

Parameter	Value
Frequency Range	88 to 858 MHz (DOCSIS / JDOCSIS) 112 to 858 MHz (EuroDOCSIS)
Carrier-to-Nose ratio at the RF input to the cable modem	30 dB
Channel bandwidth	6 MHz (DOCSIS / JDOCSIS) 8 MHz (EuroDOCSIS)

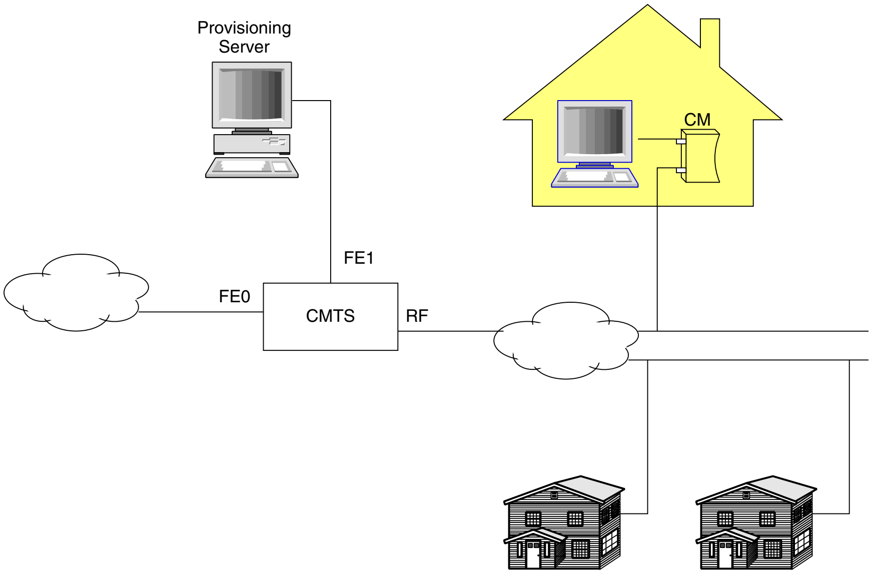
Upstream RF cable plant requirements are as follows:

Parameter	Value
Frequency Range	5 to 42 MHz (DOCSIS) 5 to 65 MHz (EuroDOCSIS / JDOCSIS)
Carrier-to-noise ratio at the RF input to the C3	At least 10 dB

Parameter	Value
Channel Bandwidth	200 KHz, 400 KHz, 800 KHz, 1600 KHz, 3200 KHz, 6400 KHz

CATV System Connections

The C3 transmitter output is the downstream RF connection (head-end to subscriber). The receiver inputs (subscriber to head end) are the upstream RF connections. There are 2 upstream connections per upstream receiver module with a maximum of 6 upstream connections per CMTS.



Unpacking the CMTS

The carton in which the Cadant C3 CMTS is shipped is specifically designed to protect the equipment from damage. Save all shipping materials in case the product needs to be returned to the manufacturer for repair or upgrade.

Unpack the equipment carefully to ensure that no damage is done and none of the contents is lost.

Package Contents

The Cadant C3 package should contain the following items:

- Cadant C3 CMTS
- Rack mounting “ears” and mounting screws
- Power cord
- Serial console cable
- Safety and Quick Start guides

If any of these items are missing, please contact your ARRIS service representative.

Action

After unpacking the equipment, but before powering it up the first time, read this manual in its entirety, then perform a visual inspection of the equipment as follows:

1. Look for the following potential problems:
    - Physical damage to the chassis or components
    - Loose connectors
    - Loose or missing hardware
    - Loose wires and power connections
2. If any of the above are found, do not attempt to power on the CMTS. Contact your local service representative for instructions.

Mounting the CMTS

The C3 CMTS is 1.75 in. (4.4 cm) high and is suitable for mounting in a standard 19 in. (48.3 cm) relay rack.

> ***Note:*** Install the CMTS in a restricted access location.

Environmental requirements

Installation of the equipment in a rack should not restrict airflow where marked on the top of the C3 case. In particular, provide adequate side clearance.

Mount the C3 properly to prevent uneven mechanical loading on the chassis. Improper mounting can cause premature failure and potentially hazardous conditions.

When installed in a closed or multi-unit rack assembly, the operating temperature inside the rack environment may be higher than ambient temperature. Ideally, you should install the C3 in an environment where the ambient temperatures remains below 40° Celsius.

Action

Follow these steps to mount the CMTS in a 19-inch rack.

1 Install one rack mounting bracket on each side of the CMTS so that the two-hole side is closest to the front of the CMTS and the brackets protrude away from the CMTS. Use four screws to fasten each bracket to the CMTS.

CAUTION
Heavy load
The CMTS weighs approximately 22 lbs (10 Kg). If necessary, have a second person hold the CMTS while mounting it to the rack.

2 Mount the CMTS in the rack and secure it using two screws on each side.

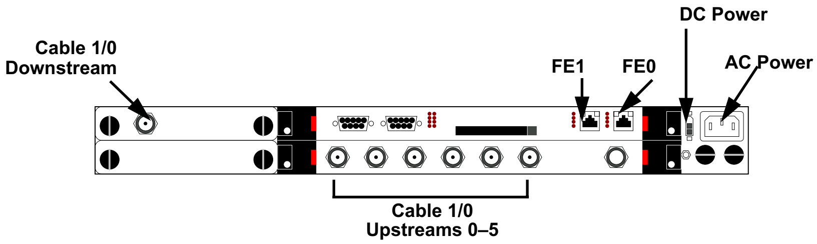
Connecting Cables

Use this procedure to connect RF, data, and power cables to the CMTS.

Depending on the configuration ordered, the C3 may have 2, 4, or 6 upstreams.

CMTS Rear View

Refer to the following figure to locate the cable ports.



Action

Follow these steps to connect cables to the CMTS.

1 Connect the upstream cable from your plant to the appropriate upstream ports. The upstream ports are located on the lower board, and are numbered left to right as viewed from the rear.

Note: Connect all RF ports to SELV circuits (meeting the requirements of SELV as defined in UL60950). Your headend must provide suitable protection between the RF ports and the CATV outside plant.

2 Connect the downstream cable to the downstream port (the F-connector located at the upper left).

3 Connect a PC to the serial connector (male DB9 connector on the upper interface module). The pin-out for this connector is designed to function with a PC when used with a straight-through cable, and is shown in the following table. The serial port operates at 38,400 bps with 8 data bits, 1 stop bit, and no parity bit.

Pin	Signal
1	Data Carrier Detect (DCD)
2	Receive Data (RD)
3	Transmit Data (TD)
4	Data Terminal Ready (DTR)
5	Ground (GND)
6	Data Set Ready (DSR)

Pin	Signal
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Unused

4 (optional) Connect an Ethernet cable between the FE1 port and the network manager.

5 Connect an Ethernet cable between the FE0 port and the network bridge or router.

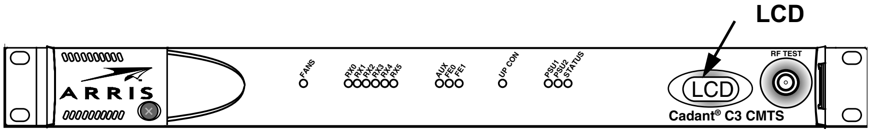
6 Make the power connection as follows:

- If using AC power, connect the power cord to the input socket in the upper right (above the fuses).
- If using DC power, connect the supplied DC power cable to the small white connector to the immediate left of the AC input connector.

Note: When DC powering, the chassis should be earthed to the rack using the supplied M4 earthing stud as detailed in “Earthing” on page 2-2.

7 Apply power to the CMTS.

The cooling fans should start to turn, and the CMTS should display initial startup messages on the LCD screen on the front panel. The following figure shows the location of the LCD.



Initial Configuration

The following sequence can be used to start up the ARRIS Cadant C3. This startup sequence assumes an “out of the box” initial condition.

Prerequisites

The following items must be set up before configuring the CMTS:

- An external DHCP server must be running.
- TFTP service must be configured in one of the following ways:
  - — An external TFTP server must contain the cable modem configuration file specified by the DHCP server. (This procedure assumes an external TFTP server.)
  - — The internal C3 TFTP server must be configured, and the cable modem configuration file stored in the configured root directory.

Optional Items

The following items are optional for the initial configuration, but may be required for normal operation:

- A ToD server is available for the cable modem.
- An NTP server is available for the CMTS.
- A Syslog server is available.

An external TFTP server is optional, since the C3 has a built-in TFTP server. If you prefer not to use the internal TFTP server, then an external TFTP server is necessary.

Initial Boot Parameters

Required boot parameters depend on how the C3 loads its software image.

If the software image is on…	**Required boot parameters are…**
the C3 flash disk	none
an external TFTP server	• booting interface (see below) • initial IP address of the booting interface • default gateway IP address to the TFTP server • the 802.1Q VLAN ID if booting over an 802.1Q VLAN encoded backbone is required

The choice of the booting interface (**fa0/0** or **fa0/1**) also pre-defines certain bridging behavior of the CMTS. You can reconfigure this behavior, but from a factory default condition before the system loads it's code for the first time (or no startup-configuration on the compact flash disk):

- Selecting **fa0/0** configures "in-band" behavior. All cable modem and CPE traffic is directed to **fa0/0**; you can use either Ethernet port for managing the CMTS.
- Selecting **fa0/1** configures "out-of-band" behavior. All CPE traffic is directed to **fa0/0**. All cable modem traffic is directed to **fa0/1**. You can use either Ethernet port for managing the CMTS.

Factory Default Network Settings

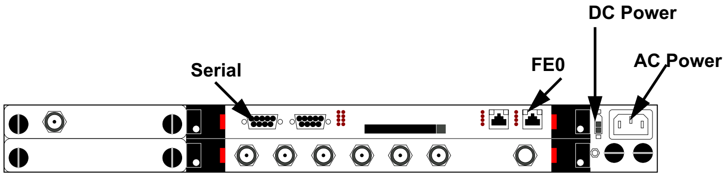
Factory default network settings are:

- IP address is one of:
  - — 10.1.127.120
  - — 10.1.127.121
  - — 10.1.127.122
  - — 10.1.127.123
- Subnet mask: 255.255.128.0
- Gateway address:10.1.0.3

See Appendix C for a complete list of factory default settings.

Rear Panel Connectors

Refer to the following diagram when performing this procedure.



Action

Perform the following tasks in the order shown.

Preparing the Connections

1 Connect the power cable to the CMTS. Do not power up yet.

2 Connect the RS232 serial cable to the serial port and connect the other end to a terminal (or PC with a terminal emulation program).

3 Start the console application and set the console configuration to:

- Port: Com1/Com2, depending on your connection
- Baud rate: 38400
- Data: 8 bits
- Parity: None
- Stop bit: 1
- Flow control: None

Verifying Proper Startup

Follow these steps to start the C3 CMTS for the first time.

1 Power on the CMTS and verify that the following status LEDs on the front panel are illuminated green:

- FANS
- PSU1
- PSU2 (if second power supply is installed)
- Status

2 Verify that the FE0 and FE1 ports on the back of the CMTS have illuminated green Link LEDs (for the port that is being used).

3 Wait for the message “Press any key to stop auto-boot...” to appear on the console, then press any key to stop auto booting before the count reaches 0.

Note: Auto booting continues after two seconds.

4 At prompt, type **help** or **?** and press ↵ to view the different commands available for boot options.

The first commands you see are user level commands.

```
CMTS>?
----------------------------------------------------------------
Command          Description
----------------------------------------------------------------
```

```
boot            Boot the CMTS using current boot parameters
bootShow        Display current boot parameters
enable          Enable Supervisor/Factory Level
sysShow         Show system configuration
timeShow        Displays current Date and Time from RTC
dir             Show directory of Compact Flash
vlevel          Set Verbosity Level
reboot          Reboot
help            Display general help or help about a command
?               Display general help or help about a command
@               Boot the CMTS using current boot parameters
>
```

Setting Boot Parameters

1 Enter privileged mode using the **enable** command to change the boot parameters. The first time you enter this mode, there is no password set and you can enter with no password. Use the **setpwd** command if a password is required in the future.

Several more commands are now available. Type ***?*** *to see the entire list.*

```
>enable
No supervisor level password set yet
Use "setpwd" command to set password
Supervisor level enabled
>?
----------------------------------------------------------------
Command         Description
----------------------------------------------------------------
boot            Boot the CMTS using current boot parameters
bootShow        Display current boot parameters
bootCfg         Configure the boot parameters
cf              Select Compact Flash for booting
tftp            Select TFTP for booting
wan             Select FA0/0(WAN) port for network access
mgmt            Select FA0/1(MGMT) port for network access
enable          Enable Supervisor/Factory Level
disable         Disable Supervisor/Factory Level
sysShow         Show system configuration
setTime         Set time in RTC
setDate         Set Date in RTC
timeShow        Displays current Date and Time from RTC
dir             Show direcory of Compact Flash
setpwd          Set password
vlevel          Set Verbosity Level
setVlanId       Set the VLAN tag to be used
vlanEnable      Enable VLAN tagging/stripping as set by setVlanId
vlanDisable     Disable VLAN tagging/stripping
reboot          Reboot
```

```
help            Display general help or help about a command
?               Display general help or help about a command
@               Boot the CMTS using current boot parameters
>
```

2 Decide what Ethernet interface to use for network access, using the commands **wan** (to select FE0) or **mgmt** (to select FE1).

*The **bootShow** command displays the selected interface as the “Network port” as shown in the next step.*

Most CLI commands refer to the FE0 port as **fastethernet 0/0.0** and the FE1 port as **fastethernet 0/1.0**.

If the CMTS has been booting from one interface and you change this interface using the above commands, you need to power cycle the CMTS for the change to take effect.

3 Enter **bootShow** to view the current boot options. (Note that the CMTS does not show the TFTP server IP address unless BootCfg is selected as following).

A listing similar to the following displays:

```
CMTS>bootShow
*** Current Boot Parameters ***
Boot from           : Compact Flash
Boot file           : C:\2.0.3.12.bin
CMTS IP Address     : 10.1.127.121
CMTS subnet mask    : ffff7f00
Gateway Address     : 10.1.0.3
CMTS Name           : CMTS
Network port        : WAN
Vlan Tagging        : Disabled
```

4 If the C3 is to be managed over an 802.1Q VLAN, make the VLAN assignment so that remote management systems can communicate with the C3 during the boot process. This is also required if the C3 is configured to boot using TFTP, since the TFTP transfer might use the VLAN. Use the **vlanEnable** and **setVlanId** commands to set up the VLAN.

```
CMTS>vlanEnable
CMTS>setVlanId 1
CMTS>bootShow
*** Current Boot Parameters ***
Boot from           : Compact Flash
Boot file           : C:\2.0.3.12.bin
CMTS IP Address     : 10.1.127.121
CMTS subnet mask    : ffff7f00
```

```
Gateway Address     : 10.1.0.3
CMTS Name           : CMTS
Network port        : WAN
Vlan Tagging        : Enabled
Vlan Id             : 1 (0x1)
C3>
```

5 To change the above list of boot options, enter **bootCfg** at the command prompt. You can change the boot parameters one at a time. Enter the new value for each parameter in turn to modify them. Then enter **bootShow** to review the changes. Set the IP address for the ARRIS Cadant C3 to suit your network.

```
>bootCfg

Options:
*[ 1]  Boot from TFTP
 [ 2]  Boot from Compact Flash
Select desired option : [ 2]
Application Image path : [ C:\2.0.3.12.bin]
CMTS Ip Address : [ 10.1.127.121]
CMTS Subnet Mask : [ 255.255.128.0]
TFTP Server Ip Address : []
Gateway Ip Address : [ 10.1.0.3]
Saving in non-volatile storage

>>
```

“Application Image path” is the name of the file and the file path if stored locally on the compact flash disk that contains the code image to be loaded. Note that the drive letter C is in UPPER CASE.

“Gateway Ip Address” is the IP address of the default router on the backbone network. The C3 uses this IP address for TFTP server booting and for the running configuration.

6 Once the boot parameters have been modified as required, boot the system by entering **@** at the prompt.

Once the system is booted, the serial port supports the CLI. When this is the first time the ARRIS Cadant C3 has been powered up, the CMTS automatically creates all of the required run time files from the specified image file.

The CMTS loads the image file and comes online.

The following output is representative of that generated on the console screen during boot and initialization.

```
*** Current Boot Parameters ***
```

```
Boot from            : Compact Flash
Boot file            : C:\3.0.1.17.bin
CMTS IP Address      : 10.1.127.121
CMTS subnet mask     : ffff7f00
Gateway Address      : 10.1.0.3
CMTS Name            : CMTS
Network port         : WAN
Vlan Tagging         : Disabled
Attached TCP/IP interface to sbe0.
Attaching network interface lo0... done.
.
.
.
etc
.
.
.

!    No CLI accounts - Telnet is disabled
!    Please configure a login account with the "cli account"
command
Arris CMTS

C3>
```

Configuring an Initial CLI Account

You must create at least one CLI account before the CMTS allows telnet access. Follow these steps to create a CLI account.

1 If you have not done so already, type **enable** to enter privileged mode.

The prompt changes to a # symbol.

2 Enter the following commands to create an account:

C3# **configure terminal** ↵

C3(config)# **cli account** {acctname} **password** {passwd }↵

The CMTS creates the account with the specified name and password.

3 Enter the following command to give privileged (enable) access to the account:

C3(config)# **cli account** {acctname} **enable-password** {enapasswd} ↵

C3(config)# **exit** ↵

Note: The login password and enable password may be the same if you prefer.

Configuring IP Networking

The C3 applies the CMTS IP address configured in the boot parameters to the fastethernet interface selected as the boot interface, and to the cable interface when booting from the default configuration (or when no startup-configuration file is available). If these settings are not suitable, use this procedure to specify the IP address information required for normal C3 operation.

Configuration Options

The C3 CMTS supports two configuration options:

- bridging (no IP routing) mode—see Chapter 3
- IP routing mode—see Chapter 5

Default Bridge Groups

Depending on the boot interface you chose in “Setting Boot Parameters” on page 2-15, the C3 pre-configures two bridge groups. See “Default Bridge Operation” on page 3-6 for a description of the initial configuration.

Action

Perform one of the following tasks:

Task	Page
Configuring Bridging Mode	2-19
Configuring IP Routing Mode......................................	2-21

Configuring Bridging Mode

Follow these steps to configure a different default route.

1 Log into the CMTS.

2 Enter one of the following groups of commands:

a To assign the management IP address to the fastethernet 0/0.0 (FE0) primary sub-interface, enter the following commands:

```
C3# config terminal ↵
C3(config)# interface fastethernet 0/0↵
C3(config-if)# ip address {mgmt-ip-addr} {mask}↵
C3(config-if)# exit ↵
C3(config)# exit ↵
C3# copy running-config startup-config ↵
```

b To assign the management IP address to the fastethernet 0/1.0 (FE1) primary sub-interface, enter the following commands:

```
C3# config terminal ↵
C3(config)# interface fastethernet 0/1 ↵
C3(config-if)# ip address {mgmt-ip-addr} {mask}} ↵
C3(config-if)# exit ↵
C3(config)# exit ↵
C3# copy running-config startup-config ↵
```

3 Enter the following commands to set the default gateway IP address:

```
C3# config terminal ↵
C3(config)# ip default-gateway {gw_ip_addr}↵
C3(config)# exit ↵
C3# copy running-config startup-config ↵
```

Configuring IP Routing Mode

Follow these steps to the configure the C3 CMTS for IP routing mode:

1 If IP routing is turned on while the subinterfaces have bridge-group memberships, or a cable sub-interface has the same IP address as a fastethernet interface in the same bridge group, changing to pure IP routing is not successful. If pure IP routing with no bridge groups is required, use step **c**; otherwise, use steps **a** and **b**.

a IP routing with bridge-group memberships:

```
C3# config terminal ↵
C3(config)# ip routing ↵
```

b Configure the default route if necessary:

```
C3# config terminal ↵
C3(config)# ip route 0.0.0.0 0.0.0.0 {route}↵
```

where

Route	IP address of the default route (or route of last resort

c True IP routing, removing bridge-group memberships:

```
C3# config terminal ↵
C3(config)# interface fastethernet 0/0.0 ↵
C3(config-if)# no bridge-group ↵
C3(config-if)# interface cable 1/0.0↵
C3(config-if)# no bridge-group ↵
C3(config-if)# interface fastethernet 0/1.0↵
C3(config-if)# no bridge-group ↵
C3(config-if)# interface cable 1/0.1↵
C3(config-if)# no bridge-group ↵
C3(config-if)# exit ↵
C3(config)# exit ↵
```

2 Set the IP address of the cable interface:

```
C3(config)# interface cable 1/0.0 ↵
C3(config-if)# ip address {cbl_ip} {subnet} ↵
```

The **cbl_ip** address may not be in the same subnet as the management IP address.

3 Configure the DHCP relay (this is required for a cable modem to register when the CMTS is in IP routing mode):

```
C3(config-if)# ip dhcp relay ↵
```

4 Cable helper address is mandatory for IP routing cable sub-interfaces that are running DHCP relay.

```
C3(interface)# cable helper-address {ipaddr} ↵
C3(interface)# exit ↵
```

5 Enter the following commands to save the routing configuration:

```
C3(config)# exit ↵
C3# copy running-config startup-config ↵
```

Configuring the Cable Interfaces

Use this procedure to configure and connect the cable upstreams and downstream.

Appendix B shows some example configurations.

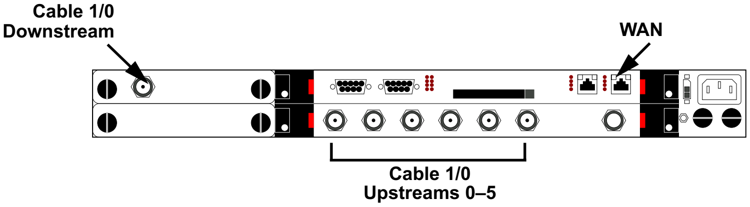
Appendix C shows the factory default configuration. The factory default configuration has the downstream in a shutdown condition so the C3 is in a passive state by default.

Requirements

Connect the downstream and any upstreams in use before performing this procedure.

Cable Connections

The following diagram shows the locations of the cable connections on the rear panel of the C3 CMTS.



Action

Perform the following tasks in the order shown.

Configuring Downstream Parameters

Follow these steps to configure the downstream cable interface.

1. Connect a PC to the CMTS, using either the serial port or the Ethernet interface (telnet connection).
2. Log into the CMTS.
3. Type **enable** to get into privileged mode, and then type the enable password.

4 Use the following commands to begin cable interface configuration:

```
C3# conf t ↵
C3(config)# interface cable 1/0 ↵
```

5 Set the downstream frequency (in Hz) using the following command:

```
C3(config-if)# cable downstream frequency {freq} ↵
```

Example: **cable downstream frequency 501000000**

6 Set the power level (in dBmV) using the following command:

```
C3(config-if)# cable downstream power-level {pwr} ↵
```

Set the power level to match the parameters assigned by the plant designer. Example: **cable downstream power-level 51**

7 (optional) Set the DOCSIS mode using one of the following commands:

```
C3(config-if)# cable downstream annex a ↵
C3(config-if)# cable downstream annex b ↵
C3(config-if)# cable downstream annex c ↵
```

8 (optional) Set the downstream modulation type using one of the following commands:

```
C3(config-if)# cable downstream modulation 64qam  ↵
C3(config-if)# cable downstream modulation 256qam ↵
```

9 Proceed to “Configuring Upstream Parameters” on page 2-25.

Configuring Upstream Parameters

Follow these steps to configure each upstream cable interface. The parameter **us** refers to the upstream interface ID, 0 to 5, corresponding to upstreams RX0 through RX*5* on the back of the C3 CMTS.

1 Set the upstream channel width (in Hz) using the following command:

```
C3(config-if)# cable upstream {us} channel width {width} ↵
```

The channel width specified must be a DOCSIS-standard upstream channel width.

ATDMA: **6400000** (6.4 MHz)

ATDMA and TDMA: **3200000** (3.2 MHz), **1600000** (1.6 MHz), **800000** (800 KHz), **400000** (400 KHz), or **200000** (200 KHz).

Example: **cable upstream 2 channel width 3200000**

2 Set the upstream channel frequency (in Hz) using the following command:

```
C3(config-if)# cable upstream {us} frequency {freq} ↵
```

The valid frequency range is **5000000** (5 MHz) to **42000000** (42 MHz) for North American DOCSIS, and **5000000** (5 MHz) to **65000000** (65 MHz) for EuroDOCSIS.

Example: **cable upstream 2 frequency 25000000**

3 (optional) Set the upstream channel modulation using one of the following commands:

a Specify a QPSK template, suitable for TDMA or TDMA and ATDMA channels:

```
C3(config-if)# cable modulation-profile {n} qpsk ↵
```

b Specify a 16QAM template, suitable for TDMA or TDMA and ATDMA channels:

```
C3(config-if)# cable modulation-profile {n} 16qam ↵
```

c Specify a mixed template using QPSK for ranging/request, 16QAM for data, 64QAM for advanced-PHY data, suitable for TDMA or TDMA and ATDMA channels:

```
C3(config-if)# cable modulation-profile {n} mix ↵
```

d Specify a template using QPSK for ranging/request, 64QAM for advanced-PHY data, suitable for ATDMA channels:

```
C3(config-if)# cable modulation-profile {n} advanced-phy ↵
```

Where *n* is a modulation profile index, **0** to **5**.

4 Assign the modulation profile to an upstream using the following command:

```
C3(config-if)# cable upstream {us} modulation-profile {n} ↵
```

Where *n* is a modulation profile index, **0** to **5**.

The factory default modulation profile for each upstream is profile 1. This profile uses QPSK and is the safest profile to use to get modems online.

5 Set the input power level (the target receive power set during the DOCSIS ranging process) using the following command:

```
C3(config-if)# cable upstream {us} power level {power} ↵
```

The valid power range depends on the channel width; the range **-4** to **14** is valid for all channel widths. See “cable upstream power-level” on page 6-141 for individual ranges.

Example: **cable upstream 2 power level 0**

6 Repeat steps 1 through 5 for each upstream that you need to configure.

7 Proceed to “Enabling the Interfaces.”

Enabling the Interfaces

Follow these steps to enable the cable interfaces.

1 Enable an upstream cable interface using the following command:

```
C3(config-if)# no shutdown ↵
```

Repeat this command for each configured upstream.

2 Enable the downstream cable interface using the following command:

```
C3(config-if)# no cable downstream shutdown ↵
```

The CMTS is now ready to acquire and register cable modems. To display the current CMTS configuration, use the **show running-config** command.

3 Bridge operation

The C3 CMTS supports IP bridging and routing modes of operation. This chapter describes bridging mode.

For more information, see:

- Chapter 4, “Providing Multiple ISP Access,” for information about using bridge groups to separate traffic and provide cable modem access to multiple ISPs.
- Chapter 5, “Layer 3 operation,” for information about the C3’s optional IP routing mode.

Terms and Abbreviations

The following are terms and abbreviations used in this chapter.

booting interface
The Fast Ethernet interface specified in the boot options. Use the **wan** command to specify fastethernet 0/0, or **mgmt** to specify fastethernet 0/1.

bridge binding
Bridge binding maps a sub-interface *A* with VLAN tag *a* to a sub-interface *B* with VLAN tag *b*; packets with tag *a* arriving on sub-interface *A* are immediately bridged to sub-interface *B* with tag *b*, and vice-versa. No other layer 2 bridging rules are followed.

bridge group
A group of sub-interfaces that may forward (bridge) packets to other sub-interfaces in the group. There is no interaction between bridge groups at the MAC level.

default cm subinterface
A designated sub-interface used for cable modem traffic until the cable modem receives an IP address from a DHCP server.

default cpe sub-interface

A designated sub-interface, used as a source sub-interface for CPE traffic when it has no VLAN tag or explicit mapping (using the **map-cpes** command).

native tagging

Cisco routing nomenclature; sub-interfaces using native tagging do not actually tag packets transmitted from that sub-interface, but the tag number is still associated with the sub-interface for internal processing purposes.

routing sub-interface

A sub-interface that supports layer 3 routing. The default sub-interface behavior is layer 2 bridging.

sub-interface

A logical subdivision of a physical interface. The C3 supports up to 64 sub-interfaces per physical interface.

VLAN tag

The VLAN ID, used to associate a cable modem or CPE with a sub-interface. The tag can be specified either in 802.1Q VLAN encapsulated packets; or in native mode, in the cable modem's VSE.

VSE

Abbreviation for Vendor-Specific Encoding. The VSE is a TLV, stored in the cable modem configuration file, that specifies the VLAN ID used to associate the cable modem's CPE with a sub-interface.

Bridging Features

The factory default operating mode of the C3 is bridging mode.

In general, normal bridging operation should not be assumed.

- In no configuration does bridging occur between the two Fast Ethernet interfaces.
- Bridging between the FastEthernet interfaces and the cable interfaces is controlled by:
  - — the selection of the boot option network interface when no startup-configuration file exists
  - — the selection of the boot option network interface when upgrading from release 2.0 to release 3.0 software
  - — but is primarily controlled and always above is over-ridden by the presence of any existing startup-configuration file and the configuration specified therein
- IP forwarding occurs even though the C3 is running in bridging mode.
- IP forwarding between bridge groups is turned off by default for security reasons.

  IP forwarding between bridge groups may be turned on using the command **ip bg-to-bg-routing** in the interface specification.
- Static routes may be defined using the **ip route** command for:
  - — C3 management traffic
  - — the DHCP relay agent
  - — IP forwarding between bridge groups (using **ip bg-to-bg-routing**)

  Note: In bridging mode, other cable modem and CPE traffic is transparent and static routes do not apply.
- Define a default gateway for the C3 using the **ip default-gateway *x.x.x.x*** command from the CLI. A default gateway has the same purposes and restrictions as a static route.

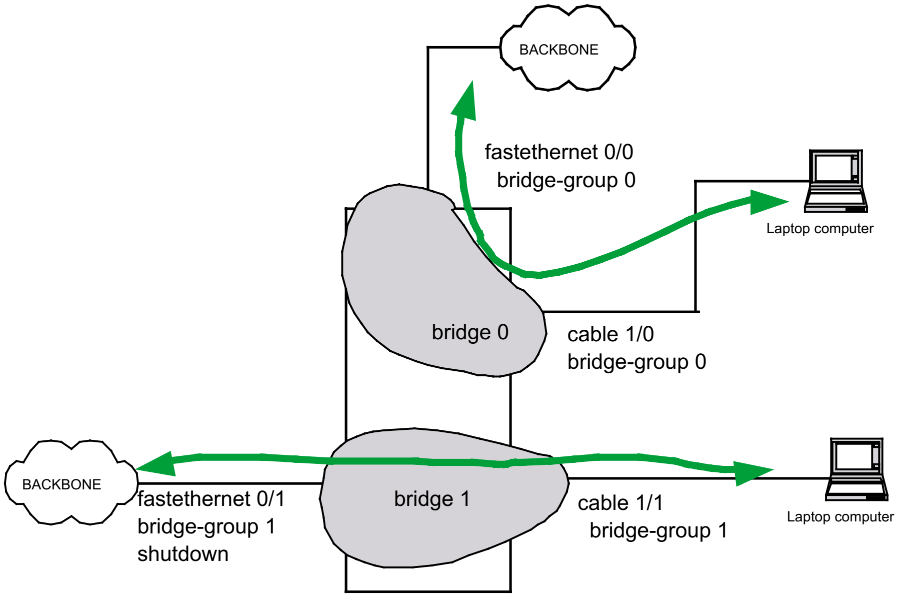
Bridge Concepts

Bridge Groups

Bridge groups provide the ability to operate self contained and separate MAC domains in one physical device.

A bridge group is defined as a group of interfaces attached to a layer 2 bridge or a common broadcast domain.

Example:



When the C3 runs in bridging mode, there is no interaction between bridge groups at the MAC level or layer 2 level—whether by ARP or any other protocol.

The problem with this concept is that although there are two physical FastEthernet interfaces, allowing each to be assigned to a separate bridge group, there is only one physical cable interface.

This issue is solved by the use of sub-interfaces.

Sub-Interfaces

Sub-interfaces split a physical interface into multiple logical interfaces to allow more flexibility in creating bridge groups. This allows each sub-interface to have different specifications for:

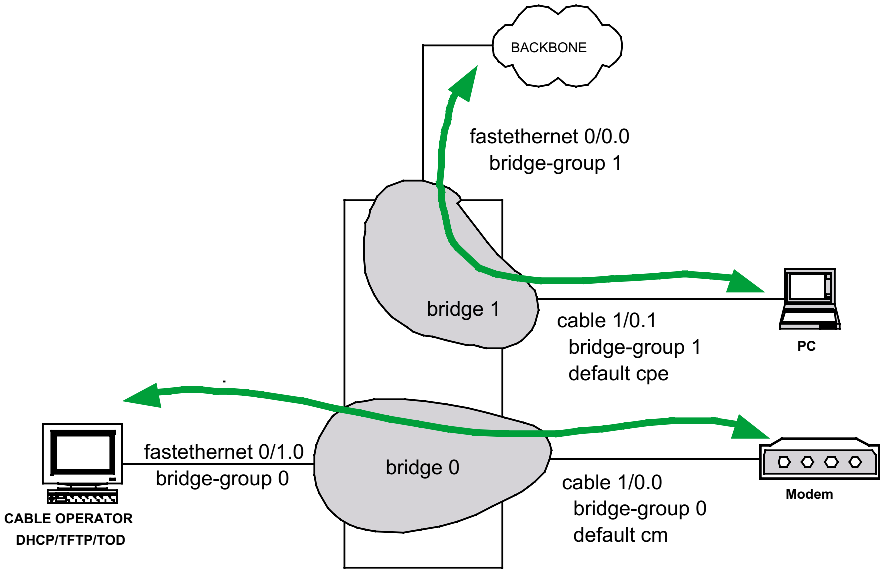
- bridge group membership
- IP addressing
- DHCP relay address provided to the DHCP server
- DHCP relay mode and helper address

- IP routing e.g. for RIP
- IGMP
- Filtering using both ACL and subscriber management
- C3 management access
- 802.1Q tagging
- other layer 3 parameters

A sub-interface is specified using a “dot” notation as follows:

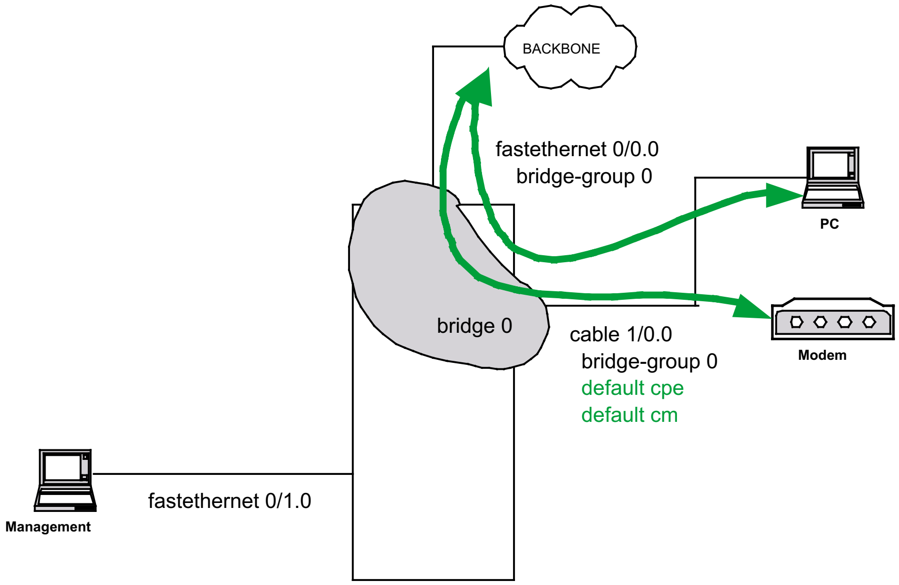
- Cable 1/0.2 is a sub-interface of the physical interface cable 1/0.
- Similarly FastEthernet 0/1.5 is a sub-interface of the FastEthernet 0/1 physical interface.

Example:



The C3 allows one sub-interface to be defined that is not a member of any defined bridge group. This interface is marked as “Management Access Only” in the “show interface” output—and as the description suggests, this interface can only be used to manage the CMTS.

Example:



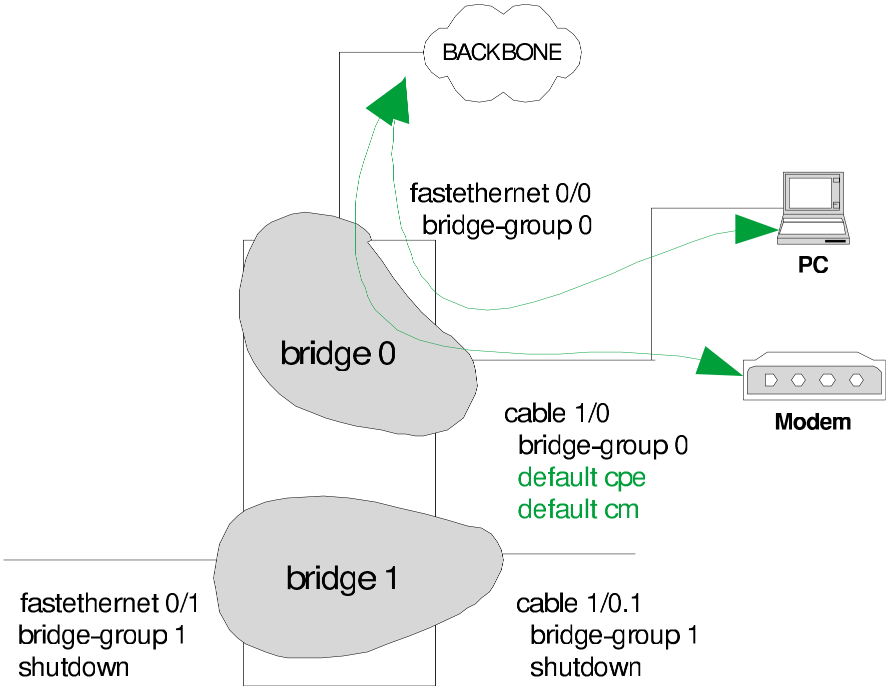
The big issue with sub-interfaces is the decision making process of how traffic is mapped from the physical interface to a sub-interface for these different specifications to have an effect. This issue is discussed later in this chapter.

Default Bridge Operation

The factory default mode of operation of the C3 is bridging mode. In this mode, the C3 has two bridge groups. Each bridge group supports up to 3 sub-interfaces. One cable sub-interface is pre-defined, but is shutdown disabling one of the bridge groups. Other sub-interfaces may be created under any physical interface subject to the above limit per bridge group.

The Additional VLAN/Bridge Group License (Product ID 713869) extends the limits to 64 bridge groups, each of which supports up to 10 sub-interfaces. Contact your ARRIS representative for ordering information and other details. See the next chapter for more details about advanced bridging, even if you are not purchasing this license.

The following figure shows the default configuration.



For more information, see:

- the CLI commands “ip default-gateway” and “ip route” for their relevance in bridging mode
- Appendix B, for sample bridging network configurations.

Selecting the Bridge Group Configuration

The above bridge group configurations may be changed:

- from the boot options using the **wan** or **mgmt** command to select the network interfaces labeled FE0 and FE1 respectively before a startup-configuration file is created on first power up. This can occur by deleting the existing startup-configuration file (using the **write erase** command) then power cycling, or the first time the C3 is powered up. In either case a default startup-configuration will be created based on the selected boot options network interface.
- by specification from the CLI after the Cadant C3 has been booted (with this configuration subsequently saved to the startup-configuration)

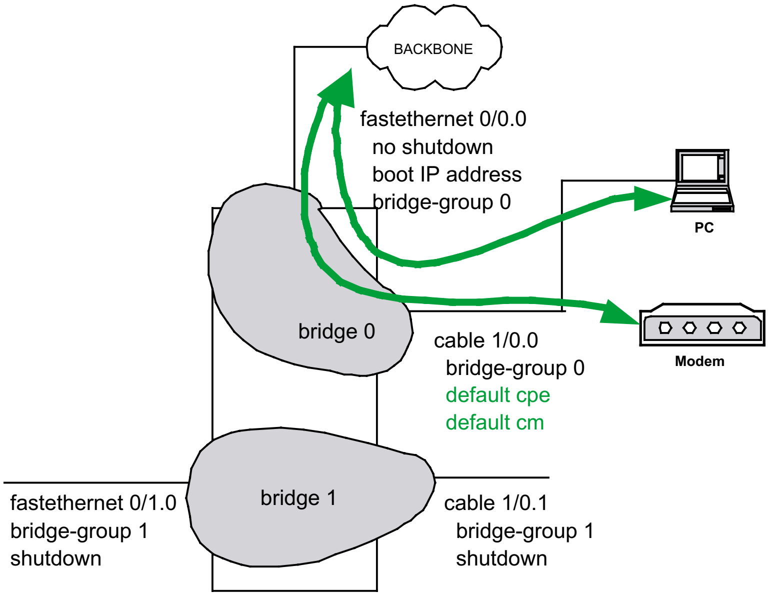
Fast Ethernet 0/0 as the Boot Options Network Interface

This is the factory default mode of operation of the C3.

In this mode, the C3:

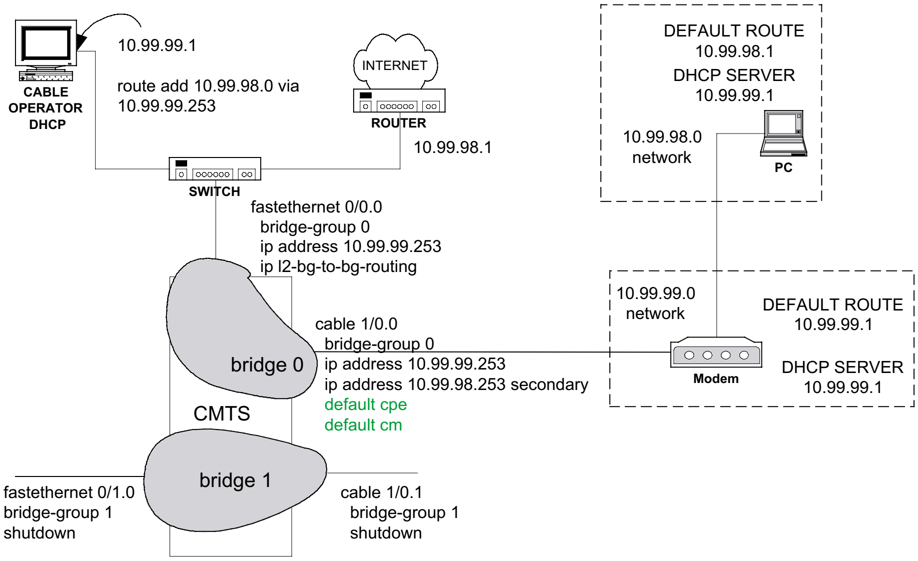
- pre-assigns interface fastethernet 0/0.0 to bridge group 0
- pre-assigns interface cable 1/0.0 to bridge group 0
- pre-assigns interface fastethernet 0/1.0 to bridge group 1, and shuts down the interface
- pre-assigns cable 1/0.1 to bridge group 1, and shuts down the interface
- sets “default cm subinterface cable 1/0.0”
- sets “default cpe subinterface cable 1/0.0”
- carries the boot option specified IP address forward into a factory default configuration as the fastethernet 0/0 IP address, and applies this IP address to the cable 1/0.0 sub-interface (this can be overwritten from the CLI).

The following diagram illustrates the default configuration.



Note: All the above settings may be changed at the CLI. For example, you can override the “management” IP address by a running-configuration specification and subsequently save it to the startup-configuration. You could also assign that IP address to the FastEthernet 0/1.0 sub-interface.

The following is an example network configuration and the CLI commands required to set it up.



```
! if the following is to be pasted to the command line
! then paste from supervisor mode
configure terminal
!
! bridges already set up from factory default
! bridge 0
! bridge 1
!
interface fastethernet 0/0.0
ip address 10.99.99.253 255.255.255.0
bridge-group 0
ip l2-bg-to-bg-routing
!
interface fastethernet 0/1.0
bridge-group 1
! no IP address required
! do not need running either
shutdown
!

interface cable 1/0.0
bridge-group 0
no shutdown
no cable upstream 0 shutdown
ip address 10.99.99.253 255.255.255.0
ip address 10.99.98.253 255.255.255.0 secondary
!
! Update giaddr with 10.99.99.253 for cable-modem
! update giaddr with 10.99.98.253 for host
ip dhcp relay
```

```
no ip dhcp relay information option
cable dhcp-giaddr policy
! unicast ALL dhcp to 10.99.99.1
cable helper-address 10.99.99.1
exit
!
interface cable 1/0.1
bridge-group 1
shutdown
!
! nothing to do here in this case
exit
exit
```

Fast Ethernet 0/1 as the Boot Options Network Interface

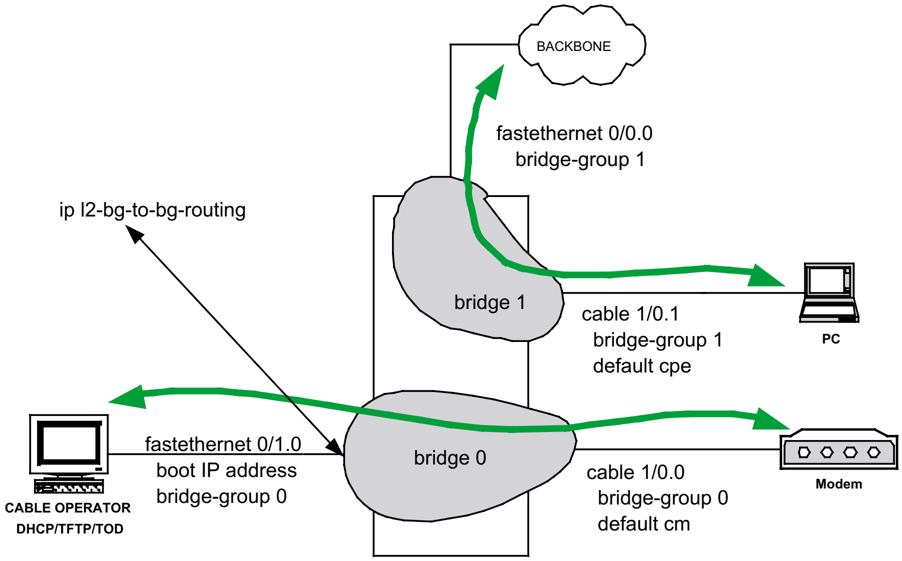
Selecting the fastethernet 0/1 interface as the boot options network interface, when there is no existing startup-configuration file, pre-assigns the bridge groups to force all cable modem traffic to the fastethernet 0/1 interface, and all CPE traffic to the fastethernet 0/0 interface. This results in “out of band” operation of the C3.

Selecting FE01 as the booting interface:

- pre-assigns interface fastethernet 0/0.0 to bridge group 1
- pre-assigns interface cable 1/0.0 to bridge group 0
- pre-assigns interface fastethernet 0/1.0 to bridge group 0
- pre-assigns cable 1/0.1 to bridge group 1
- sets “default cm subinterface cable 1/0”
- sets “default cpe subinterface cable 1/0.1”
- carries the boot option specified IP address forward into a factory default configuration as the fastethernet 0/1 IP address.

Again, all the above settings may be changed at the CLI.

The following diagram shows data flow in the C3 when fastethernet 0/1 is the boot interface.



In this example, DHCP relay must be turned on in the cable 1/0.1 sub-interface specification if CPE DHCP is to be served by a DHCP server on the fastethernet 0/1 sub-interface (MGMT port).

In addition, **ip l2-bg-to-bg-routing** must be enabled on the fastethernet 0/1.0 sub-interface for the CPE DHCP Renew to succeed. The DHCP Relay function routes the Renew from cable 1/0.1 to the fastethernet 0/1.0 sub-interface. The DHCP Renew ACK received at the fastethernet 0/1.0 sub-interface must be routed across bridge groups to cable 1/0.1; but the ACK is not destined for cable 1/0.1, so the ACK is not routed by the DHCP Relay function and fastethernet 0/1.0 must have **ip l2-bg-to-bg-routing** activated.

For more information, see the network examples in Appendix B.

Decide what is Management Traffic

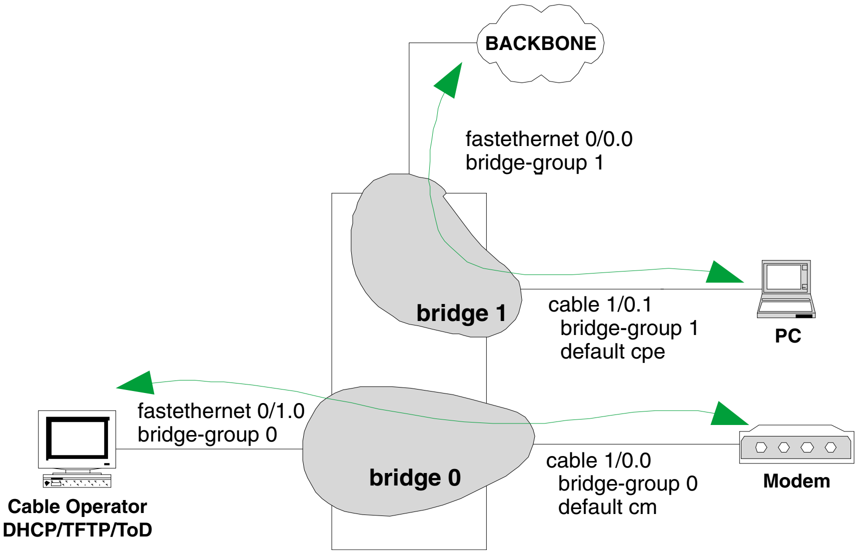
Software releases prior to v3.0 locked the user into accepting cable modem traffic as “management” traffic.

This software release allows the user to decide what is management traffic:

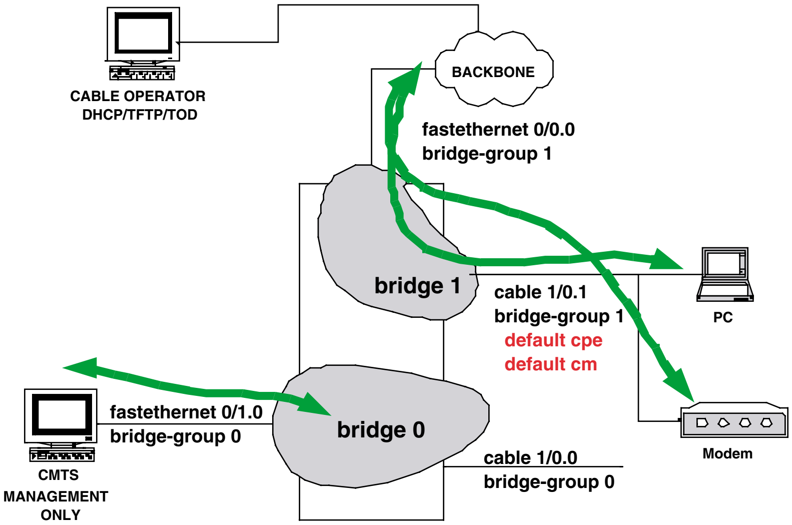
- CMTS traffic only, or
- CMTS and cable modem traffic

By re-defining the default cable sub-interface for modem traffic, modem traffic can be removed from the bridge group that contains the CMTS management traffic. This requires that the modem DHCP, TFTP, and ToD servers be present on the fastethernet 0/0 interface as in the following example.

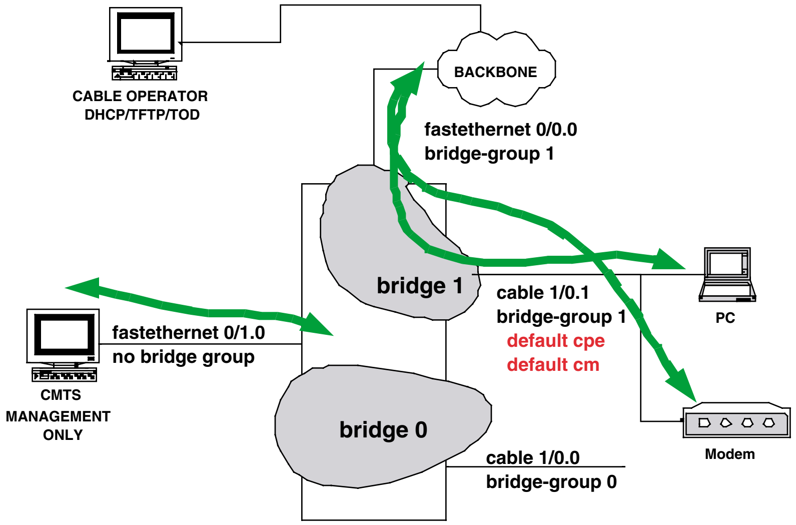
The following diagram shows the default, version 2.0-compatible, operating mode. CMTS management traffic and cable modem traffic share bridge group 0.



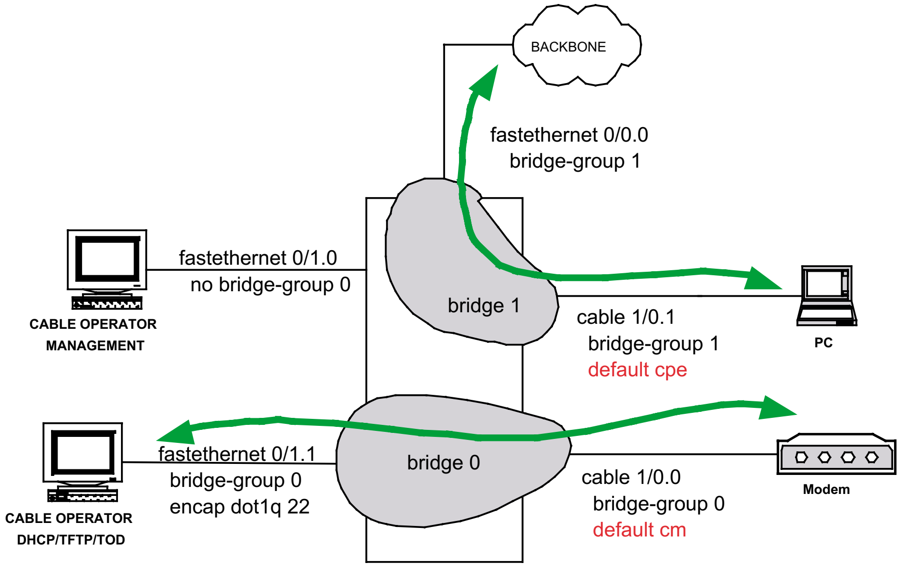
The following diagram shows bridge group 0 restricted to carrying CMTS management traffic, and bridge group 1 used for all cable modem and CPE traffic.



The following diagram shows bridge group 0 unused, and bridge group 1 used for all cable modem traffic. CMTS management traffic is restricted to a management-only sub-interface. This sub-interface is configured with the CMTS IP address and has **management access** enabled.



The final example shows CMTS management traffic on a management-only sub-interface, as before, and cable modem traffic and CPE traffic on separate bridge groups.



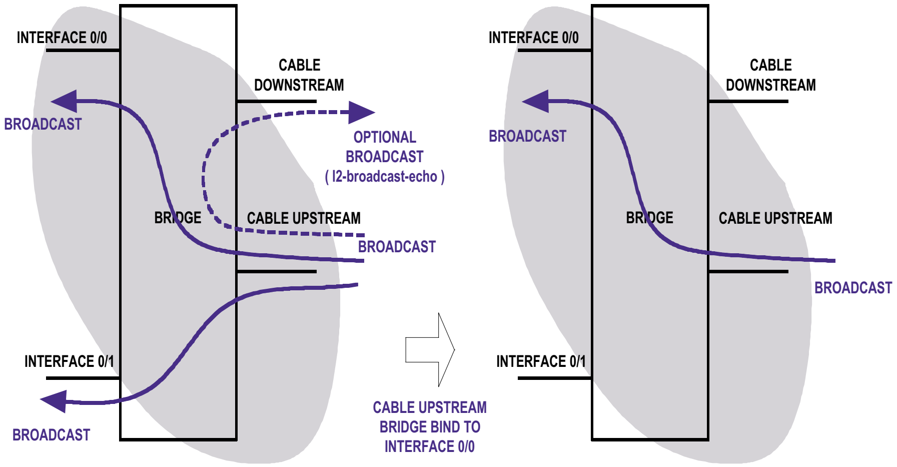
Bridge Binding

Bridge binding provides a direct link between a tagged cable sub-interface and a tagged FastEthernet sub-interface.

The cable sub-interface may use a native tag (used with VSE or **map-cpes**) or may use normal 802.1Q tagging. A FastEthernet interface must use 802.1Q tagging for bridge binding purposes.

Using a bridge bind specification can further reduce the broadcast domain. This is especially relevant in the cable interface where the downstream and upstream are treated as separate interfaces in the bridge group. A layer 2 broadcast received at the cable interface is re-broadcast on all interfaces attached to the bridge group. This includes the cable downstream interface if the command **l2-broadcast-echo** is present. This characteristic of the cable interface can be a security risk. Use of the bridge bind is one method provided in the C3 to restrict such broadcasts propagating into the cable downstream or to unwanted Ethernet interfaces.

The following diagram shows the effect of bridge binding on upstream Layer 2 broadcasts:

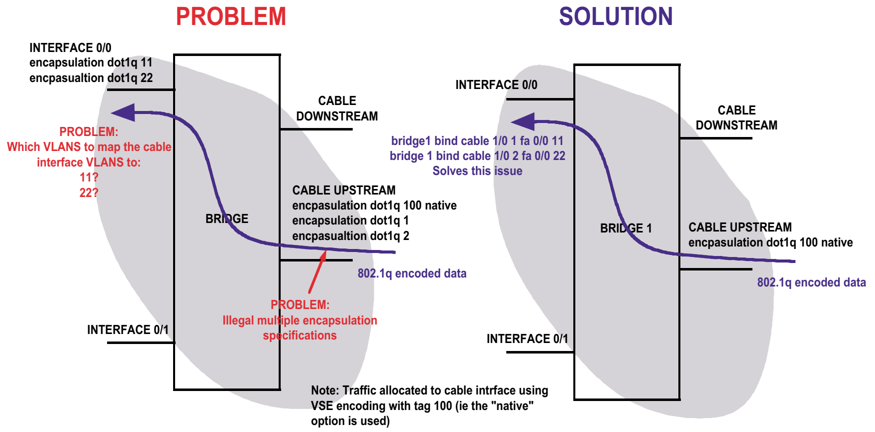


Bridge binding may be used in another way.

If all CPE traffic is allocated to a cable sub-interface (how this is done is described following), it is possible to further restrict this traffic to 802.1Q encoded traffic by specifying an encapsulation command on the cable sub-interface. This would allow a number of 802.1Q VLANs to terminate on the cable sub-interface.

In fact the multiple encapsulation commands under the cable and fastethernet interfaces are illegal and will be rejected by the CLI.

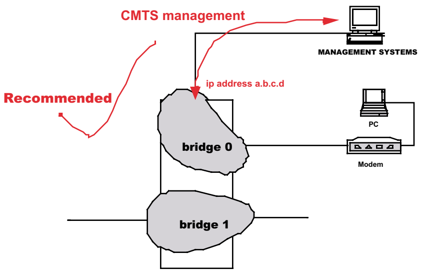
This problem is shown in the following figure. The following example shows the legal use of the **bridge bind** command to implement the same configuration as that defined as the problem in the following figure.

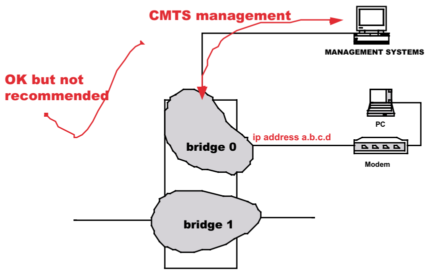


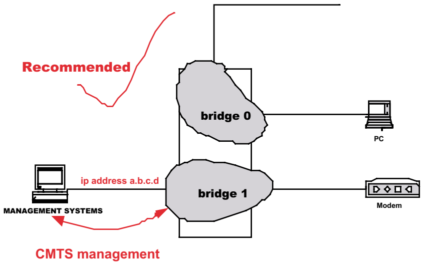
IP Addressing

A bridge does not require an IP address to operate. The C3 however can be managed over an IP network and thus must be assigned a valid IP address for management purposes.

Due to the nature of operation of a bridge, any interface in either of the two default bridges on the C3 may be assigned an IP address and this IP address may be accessed again from any interface in the same bridge group for management purposes. You can also assign the same IP address to both a cable and fastethernet sub-interface; this allows continued management access of one of the interfaces is shut down for any reason.







This “management” IP address is normally assigned from the serial console and is programmed in the startup-configuration file found on the compact flash disk.

Do not confuse the management IP address with the IP address set in the boot options. The C3 uses the IP address specified in boot options and the booting Fast Ethernet interface only if a TFTP server based boot is required—the IP address provides enough IP information to allow a TFTP server based boot to occur.

As the above diagram shows, you can assign the management IP address to a cable sub-interface. This is not recommended. If the cable interface is shutdown, you cannot manage the C3 from the network. Serial console access is not affected.

Replacing a Legacy Bridging CMTS

If the C3 is to be used in a system where only one IP address is allocated to the CMTS, and C3 DHCP relay is also required, the cable interface must have an IP address for DHCP relay to operate. In this case, in bridging mode, the cable interface can be allocated the same IP address as the “management” Fast Ethernet interface in the same bridge group.

Attaching Bridge Groups

Since a bridge group operates at the MAC layer, it can bridge IP protocols. However, the bridge group forms an isolated MAC domain and only has knowledge of devices connected to it. The bridge group can recognize IP protocols when it is attached to the C3's IP stack.

Attaching a bridge group to the IP stack requires at least one sub-interface in the bridge group to have an IP address, and for that sub-interface to be operationally up.

When a bridge group is attached, whether the C3 is configured for IP routing or bridging mode, IP packets entering the bridge group (whose MAC destination address is an interface on the C3) can now be passed to the C3's IP stack and IP-level communication between bridge groups can occur.

> ***Note:*** When running in IP routing mode, such IP forwarding is performed at wire speed. When running in bridging mode, the C3 does not support wire speed processing and such forwarding is designed to support DHCP operations only.

This communication is not always desirable, as it degrades bridge group isolation. Therefore, this function is turned off by default for every sub-interface created from the CLI. Use the sub-interface command **ip l2-bg-to-bg-routing** to allow such IP traffic to leave a bridge group and be passed to the IP stack. In some cases, this is a required step for DHCP to be successful.

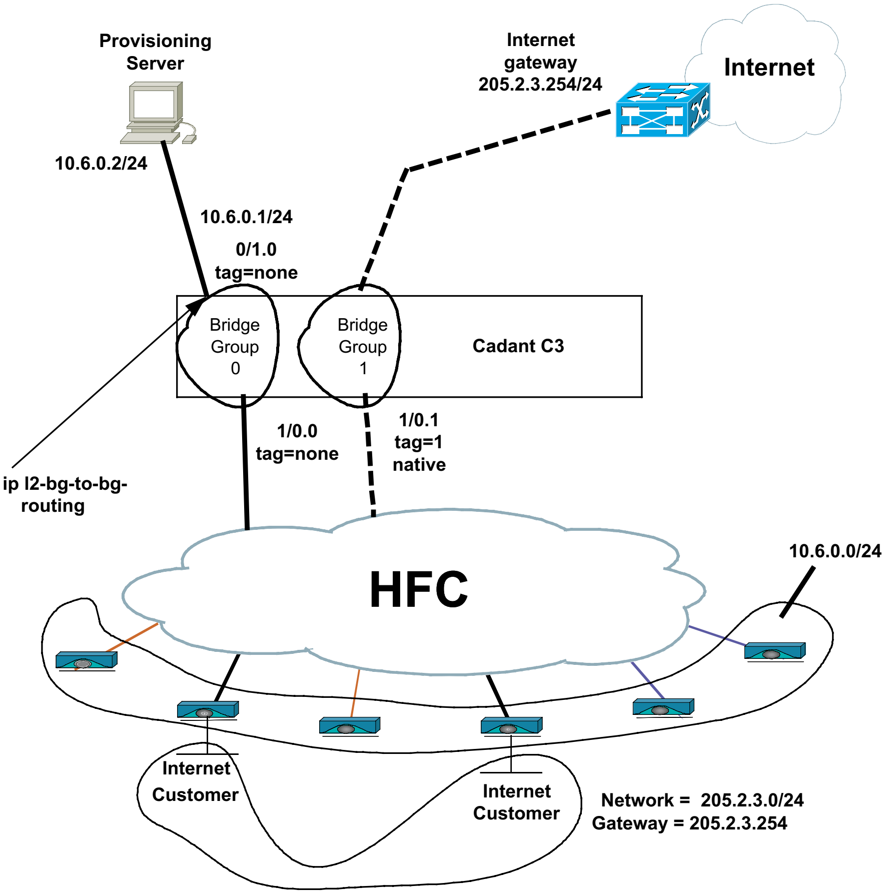
In the following example:

- modem traffic is isolated to bridge group 0—the same bridge group that the DHCP server is connected to
- modem DHCP succeeds, even if DHCP relay is not turned on

Now consider the CPE devices:

- All CPE traffic is isolated to bridge group 1
- DHCP relay must be activated on cable 1/0.1 for DHCP from the CPE to reach the DHCP server connected to fastethernet 0/1.0
- DHCP relay requires that cable 1/0.1 be given an IP address.
- The DHCP ack and offer from the DHCP server will be received at fastethernet 0/1.0
- DHCP relay will forward the offer or ack back to the relaying interface—the cable 1/0.1 sub-interface.

- The ACK to a CPE DHCP renew is not captured by the DHCP Relay function (being addressed to the CPE and not the cable 1/0.1 sub-interface) but must be forwarded across bridge groups to the CPE device. For the ACK to be forwarded across bridge groups, **ip l2-bg-to-bg-routing** again must be specified on fastethernet 0/1.0. No other sub-interface needs an **ip l2-bg-to-bg-routing** specification.



Incoming Traffic Allocation to a Sub-Interface

As detailed above, the concept of bridge groups and sub-interfaces is very powerful but hinges on how traffic arriving by a physical interface is allocated to a sub-interface by the Cadant C3.

In summary:

- Fastethernet sub-interfaces use 802.1q VLAN tags
- Cable sub-interfaces use:
  - VSE encoding
  - the **map-cpes** command
  - the **default cpe subinterface**

  If a mapped frame has an 802.1Q tag, the C3 verifies that the tag is correct for the mapped sub-interface; if the tag does not match, the C3 drops the frame.

Fastethernet Interface

802.1Q VLAN tags are used to allocate incoming packets to FastEthernet sub-interfaces with matching **encapsulation dot1q** specifications.

Only one FastEthernet sub-interface per physical interface may have no encapsulation configured. All untagged traffic is directed to this sub-interface. If a second FastEthernet sub-interface is defined with no VLAN tag, the sub-interface configuration is ignored and a CLI message warns of the incomplete configuration and informs the user which is the current untagged sub-interface.

Cable Interface

Default Mapping of CM to a Sub-Interface

If a global specification **default cm subinterface cable X/Y.Z** is present in the C3 global configuration, then all modem traffic received is mapped to the nominated cable sub-interface until the cable modem receives an IP address from DHCP and moves to its correct sub-interface. Note this is a default mapping and will be overridden by any modem IP address based mapping once the modem has an IP address.

If no default is specified, the C3 automatically assigns cable 1/0.0 as the default sub-interface.

Cable Modem IP Traffic

When a cable modem receives a DHCP Ack, the C3 inspects the assigned IP address to determine which sub-interface that the cable modem should be assigned to. The C3 maps all subsequent IP traffic from that cable modem to the designated sub-interface.

If no match can be found in any cable sub-interface, the IP packet is mapped to the default cable sub-interface.

CPE Traffic

Upstream CPE traffic may be allocated to cable sub-interfaces using:

- VSE encoding
- **map-cpes** specification
- **default cpe subinterface** specification

If a mapped frame has an 802.1Q tag, the C3 verifies that the tag is correct for the mapped sub-interface; if the tag does not match, the C3 drops the frame.

Again, one cable sub-interface may have no encapsulation specification. All other cable sub-interfaces must have an encapsulation specification in the form:

- **encapsulation dot1q X** or
- **encapsulation dot1q X native**

VSE and 802.1Q Native Tagging

The combination of native tagging and VSE encoding is one method that allows CPE traffic to be mapped to a cable sub-interface.

A cable sub-interface with native tagging means that:

- all traffic received at this interface will be internally tagged by the C3 before being passed to the bridge group the sub-interface is a member of.
- Traffic leaving the bridge group via this natively tagged sub-interface will NOT be tagged as it leaves the C3.

Contrast this behavior with the 802.1Q tagging on a FastEthernet sub-interface where all traffic leaving the C3 is tagged if the FastEthernet sub-interface has an 802.1q tag specification.

Thus native tagging is a means to identify traffic that has arrived at a particular cable sub-interface. This native tagging can also be used to map CPE traffic to a cable sub-interface.

During registration with the CMTS, all modems send a Vendor ID TLV, identifying the modem vendor to the CMTS in addition to any information received by the modem in the configuration file sent to the modem.

A cable modem configuration file may have added to it Vendor Specific Encoding (VSE) that can be used to send proprietary information to a vendor’s modems. If a modem receives such information and this information has a vendor_id that does not match that of the modem vendor, the modem ignores this information. Thus a single configuration file may contain vendor specific information for multiple vendors without any impact on modems without a matching vendor_id. This is the original purpose of this DOCSIS feature.

Regardless of whether the modem has a matching vendor_id to the configuration file specified vendor specific information or not, the modem must under DOCSIS specifications send all such received information to the CMTS during registration.

This means that the C3 receives all vendor specific information that the modem received in its configuration file.

> ***Note:*** The C3 ignores all other vendor-specific information; for example, the C3 ignores a Thomson vendor_id.

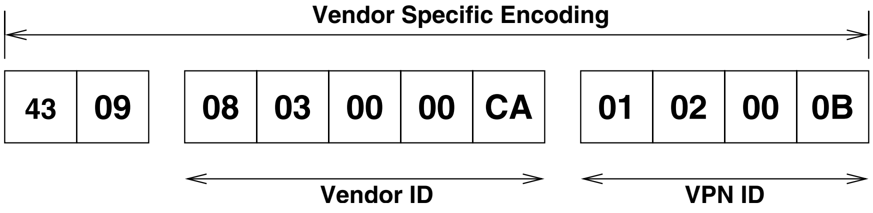
This mechanism thus provides a method to transfer information from a modem configuration file and the provisioning systems to the C3 during modem registration.

The C3 inspects all vendor specific encoding received during registration and accepts VSE information with an ARRIS vendor ID. This TLV can contain a number that identifies what cable sub-interface native tag all traffic passing through this modem is mapped to.

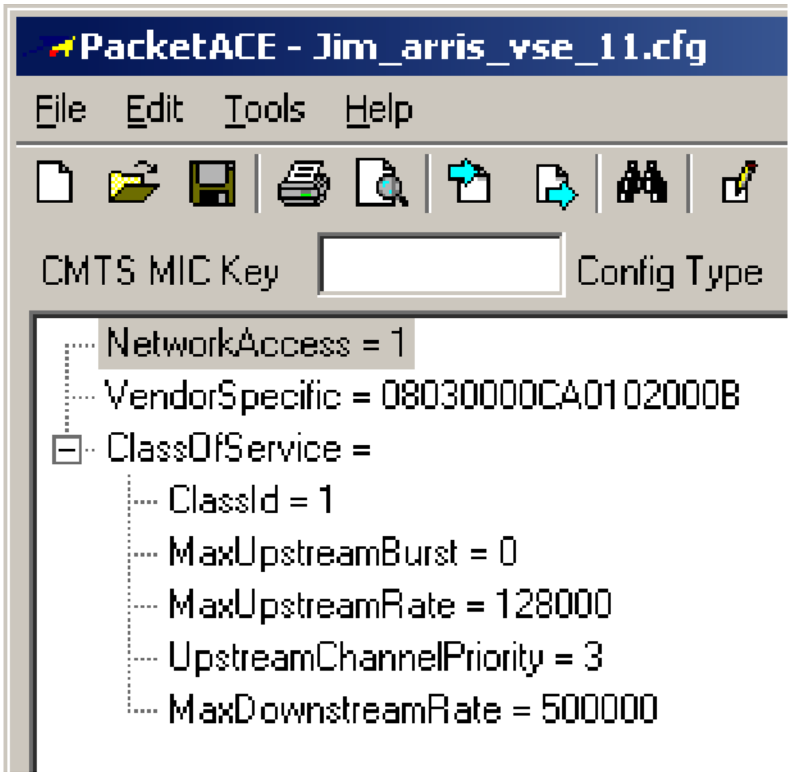
Thus all CPE traffic passing through a modem that received this configuration file can be mapped to a particular cable sub-interface.

Important: The C3 ignores all other vendor specific information; e.g. the C3 ignores a Thomson vendor_id.

The following diagram shows an example of an ARRIS VSE with a VPN ID of 000Bh (11 decimal).



The following diagram shows an example of a configuration file containing such VSE information - a VSE tag of 11 decimal is shown:



```
NetworkAccess = 1
VendorSpecific = 08030000CA0102000B
ClassOfService =
    ClassId = 1
    MaxUpstreamBurst = 0
    MaxUpstreamRate = 128000
    UpstreamChannelPriority = 3
    MaxDownstreamRate = 500000
```

If no VSE messages are received from a modem during registration, traffic from any attached CPE devices will be allocated using any **map-cpes** specification or **default cpe subinterface** specification. If no default is specified, the C3 automatically assigns cable 1/0.0 as the default sub-interface.

Example:

Let us first review quickly how standard non-DOCSIS aware DHCP servers allocate IP addresses.

DHCP servers use the giaddr IP address—the relaying IP address—to indicate from which address pool an IP address should be allocated from. It is thus important that the relaying address or the giaddr address be a meaningful address on the relaying device.

Defining cable sub-interfaces for CPE devices allows this to happen. Each cable sub-interface can have a different IP address specification with the IP address being used to populate the giaddr field as determined by the DHCP specifications of this sub-interface.

```
configure terminal
bridge 13
cable 1/0.0
```

```
! for modem only
bridge-group 0
ip address 10.99.99.1 255.255.255.0
ip DHCP relay
cable helper-address 10.0.0.1 cable-modem
cable dhcp-giaddr primary

cable 1/0.11
! for cpe with IP address
bridge-group 1
! define ip address
ip address 10.11.0.1 255.255.255.0
ip DHCP relay
cable helper-address 10.0.0.1 host
cable dhcp-giaddr primary
! for CPE traffic via modem with VSE tag = 11
encapsulation dot1q 11 native

cable 1/0.13
! for cpe layer 2 forwarding
! for CPE traffic via modem with VSE tag = 13
bridge-group 13
encapsulation dot1q 13 native
```

map-cpes

The **map-cpes** command allows re-direction of CPE traffic attached to a modem to a specified cable sub-interface.

Once a modem is allocated an IP address, the modem is mapped to any cable sub-interface that has a matching subnet. Thus if modems are allocated to different subnets, they can be mapped by the C3 to different cable sub-interfaces.

If a **map-cpes** specification is in place in the cable sub-interface that the modem is allocated to, all incoming CPE frames arriving via this modem are allocated to the specified cable sub-interface.

Example:

```
configure terminal
bridge 11
interface fastethernet 0/0.1
bridge-group 11
encapsulation dot1q 111

interface cable 1/0.0
! for modem only
bridge-group 0
ip address 10.99.99.1 255.255.255.0
ip dhcp relay
```

```
cable helper-address 10.0.0.1 cable-modem
cable dhcp-giaddr primary
map-cpe cable 1/0.11

interface cable 1/0.11
! for cpe bridging
bridge-group 11
! accept 802.1q tagged frames only
encapsulation dot1q 11
```

Default Mapping of CPE to a Sub-Interface

If a the global specification **default cpe subinterface cable X/Y.Z** is present in the Cadant C3 global configuration, the C3 maps all CPE traffic from any modem that cannot be mapped to any sub-interface to the this nominated default cable sub-interface and hence to a default cable VPN. Note this is a default mapping and is overridden by any VSE or **map-cpes** based mapping.

If no other form of mapping is used then the default mapping is cable 1/0.0 (the default cable sub-interface).

CPE 802.1Q Traffic

The C3 uses 802.1Q tags for verification and binding purposes.

If a mapped incoming frame has an 802.1Q tag, the C3 verifies that the tag is correct for the mapped sub-interface; if the tag does not match, the C3 drops the frame.

If the incoming frame has an 802.1Q header but this frame is mapped to a cable sub-interface by a **map-cpes** specification, the mapped sub-interface must have a matching 802.1Q tag for this frame to be accepted.

In either case, the C3 passes the frame to the bridge group this cable sub-interface is a member of, bridging the frame to other sub-interfaces assigned to the bridge group.

Frames bridged to fastethernet sub-interfaces are treated as follows:

- If the fastethernet sub-interface has an encapsulation specification, the C3 encodes the frame with this tag and the frame leaves the CMTS with an 802.1Q encoding.
- If the fastethernet sub-interface does not have an encapsulation specification, the C3 strips the 802.1Q header and the frame leaves the CMTS untagged.

Note that the cable interface 802.1Q tag can be different from the fastethernet interface 802.1Q tag.

Example:

```
configure terminal
bridge 11
!
fastethernet 0/0.1
bridge-group 11
encapsulation dot1q 111

cable 1/0.0
! for modem only
bridge-group 0
ip address 10.99.99.1 255.255.255.0
ip dhcp relay
cable helper address 10.0.0.1 cable-modem
cable dhcp-giaddr primary
map-cpes cable 1/0.11

cable 1/0.11
! for cpe bridging
bridge-group 11
! accept 802.1q tagged frames only
encapsulation dot1q 11
```

bridge bind

The bridge bind can be used to bind a cable sub-interface directly to a FastEthernet sub-interface as detailed earlier. A bridge-bind can also be used with VSE and 802.1Q native encoding.

The following example shows CPE traffic mapped to a cable sub-interface using VSE encoding. All traffic is bridged and VLAN tagged on exit from the bridged fastethernet sub-interface.

A series of bridge-bind specifications also adds support for 802.1Q tagging to this cable sub-interface cable 1/0.13. This facility has been used by a customer to provide tiered services inside the VPN formed by the combination of the mapping of CPE traffic to this cable sub-interface and the use of the command **encapsulation dot1q xx encrypted-multicast** to provide downstream broadcast privacy to CPE using this cable-sub-interface. See Chapter 4 for more details.

Example:

```
Bridge 0
Bridge 1
bridge 2

int fa 0/0.0
! management ip address
ip address 10.1.0.1 255.255.255.0
```

```
bridge-group 0

int fa 0/0.13
bridge-group 2
! no ip address
encapsulation dot1q 13

int cable 1/0.0
! for modem only
ip address 10.99.99.1 255.255.255.0
bridge-group 0
ip dhcp relay
cable helper-address 10.0.0.1 cable-modem
map-cpes ca 1/0.13

int cable 1/0.13
bridge-group 2
! for cpe layer 2 forwarding
encapsulation dot1q 13 native
! create VPN privacy
encapsulation dot1q 13 encrypted-multicast

exit

! all traffic ariving at cable 1/0.13
! check for tag 4, bridge to fa 0/0.13
! and tag with 44 before leaving
bridge 2 bind cable 1/0.13 4 fastethernet 0/0.13 44

! all traffic ariving at cable 1/0.13
! check for tag 5, bridge to fa 0/0.13
! and tag with 55 before leaving
bridge 2 bind cable 1/0.13 5 fastethernet 0/0.13 55
```

Traffic allocation—summary

The C3 processes incoming cable modem packets as follows:

- Before the cable modem receives an IP address, the C3 assigns all incoming packets from that cable modem to the default CM sub-interface.
- When the cable modem receives a DHCP Ack, the C3 inspects the assigned IP address and uses that to assign further cable modem packets to a sub-interface.

The C3 processes incoming CPE packets in the following order:

1 Check for modem based VSE encoding and map the traffic to a cable sub-interface with an encapsulation tag matching the VSE tag allocated to the modem; then go to step 5.

2. Check the sub-interface the attached modem is assigned to for a **map-cpes** specification; if found, map the CPE traffic to the specified cable sub-interface, then go to step 5.
3. Check for default mapping of CPE to a cable sub-interface using the **default cpe-subinterface** specification and map CPE traffic to this cable sub-interface; then go to step 5.
4. Check for CPE-based 802.1Q VLAN tagging against the mapped sub-interface VLAN specification (specified under the cable sub-interface or using a bridge-bind specification). Bridge the frame with a matching tag and drop the frame if:
   - the VLAN specification does not exist, or
   - the VLAN specification exists but does not match the frame
5. Check that the sub-interface exists and is active. If not active or does not exist then drop the data frame.

This testing is performed for modem-sourced frames and CPE-sourced frames arriving via a cable modem.

The only test above that is relevant to a cable modem is the test allowing modems to be allocated to cable sub-interfaces based on the allocated modem IP address.

Upgrading from v2.x to v3.0 Software

When version 3.0 or later software is installed on a system with a 2.0 startup-configuration file, the C3 attempts to mimic the 2.0 setup as best it can, but some human intervention is likely. This procedure describes the steps needed to finish the upgrade to version 3.0. Appendix B provides several upgrade examples.

Configuration Differences

Version 2.0 had no concept of bridge groups, and operated in either *inband* mode where fastethernet 0/1 (MGMT) is non-operational, or *out-of-band* mode where CPE traffic was bridged through fastethernet 0/0 (WAN) and CM/CMTS management traffic through fastethernet 0/ 1 (MGT).

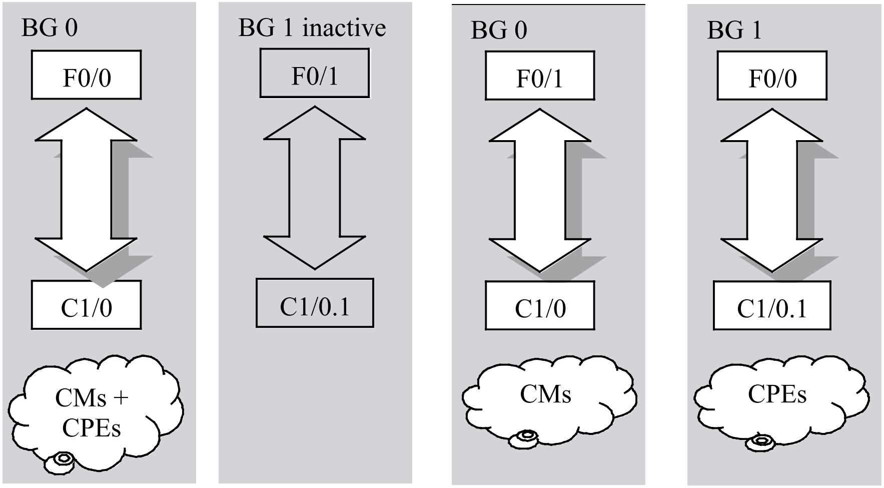
The terms “WAN” and “MGMT” are no longer used in v3.0, as either fastethernet interface can be for any purpose. The terms “inband” and “out of band” are also used sparingly in v3.0 software and the user now has complete flexibility in configuration making these terms descriptive only—there is no longer any support for the command **inband-management** in v3.0 software.

On upgrading, two bridge-groups are created. This allows the flexibility of handling cable modem traffic on one bridge group and CPE traffic on another. A management access-only sub-interface—which does not belong to any bridge group—is also allowed for CMTS management (but needs to be configured if required).

The bridge group configuration depends on whether you are upgrading from a v2.X inband or out-of-band system:

- Upgrading from 2.0 inband mode:
  - Bridge group 0 contains fastethernet 0/0.0 (WAN) and cable 1/0.0.
  - Bridge group 1 contains fastethernet 0/1.0 (MGMT) and cable 1/0.1, which are administratively down, as the bridge group is not used.
- Upgrading from 2.0 out-of-band mode:
  - Bridge group 0 is for cable modems and contains fastethernet 0/1.0 MGT and cable 1/0.0.
  - Bridge group 1 is for CPE traffic and contains fastethernet 0/0.0 WAN and cable1/0.1

— The command **default cpe subinterface cable 1/0.1** is applied. All CPEs use this sub-interface (and thus belong to bridge group 1).



2.0 inband after upgrade *2.0 out-of-band after upgrade*

The version 2.0 boot address is applied to both sub-interfaces in bridge group 0 on upgrading. Any IP addresses (including secondary specifications) for sub-interfaces in the 2.0 startup configuration are applied to the same physical interfaces in the 3.0 setup. Secondary IP addresses for cable sub-interfaces have to be manually configured (configuring IP addresses on the cable interface was not possible in the 2.0 release).

Action

Follow these steps to complete the upgraded configuration for use with version 3.0 software.

1 If you were using DHCP relay previously, you must enable it on each active cable sub-interface. The **ip dhcp relay** command was global in 2.0, and is per-cable sub-interface in 3.0. Use the following commands to enable DHCP relay:

```
conf t
interface cable 1/0.x
ip dhcp relay
```

2 The **ip default gateway** command is always commented out in 2.0 configuration files, since it was set automatically from the boot options. If the default gateway is required, add the command to the configuration.

3 If access lists applied against cable 1/0 are configured for CPE devices, then you need to reconfigure those access lists for sub-interface cable 1/0.1 if the C3 was running in out-of-band mode.

4 DHCP cable helper addresses applied to the cable interface in both version 2.0 and version 3.0 may have to be applied to other cable sub-interfaces if necessary. For example, if the C3 was running in out-of-band mode, apply all common helper addresses to cable 1/0.1, plus all helper addresses marked “host.” The cable 1/0.0 sub-interface should retain all common helper addresses and all those marked “cable-modem.” For example:

```
cable helper-address 4.5.6.6
! should appear on C1/0.0 and C1/0.1
cable helper-address 4.5.6.7 cable-modem
! c1/0.0 only (CMs)
cable helper-address 4.5.6.8 host
! c1/0.1 only (CPEs)
```

5 In version 3.0 software, dot1q encapsulation is required to differentiate cable sub-interfaces, even if VLAN tags are not used. The upgrade-generated C1/0.1 sub-interface is encapsulated using the **encapsulation dot1q 1 native** command. The upgrade-generated C1/0.0 sub-interface remains untagged.

6 The old **cable vpn cmts X** and **cable vpn cm Y** VLAN tagging commands are not supported in 3.0. To support similar functionality, configure a CMTS management-only sub-interface with the IP address of the CMTS and the appropriate VLAN tag.

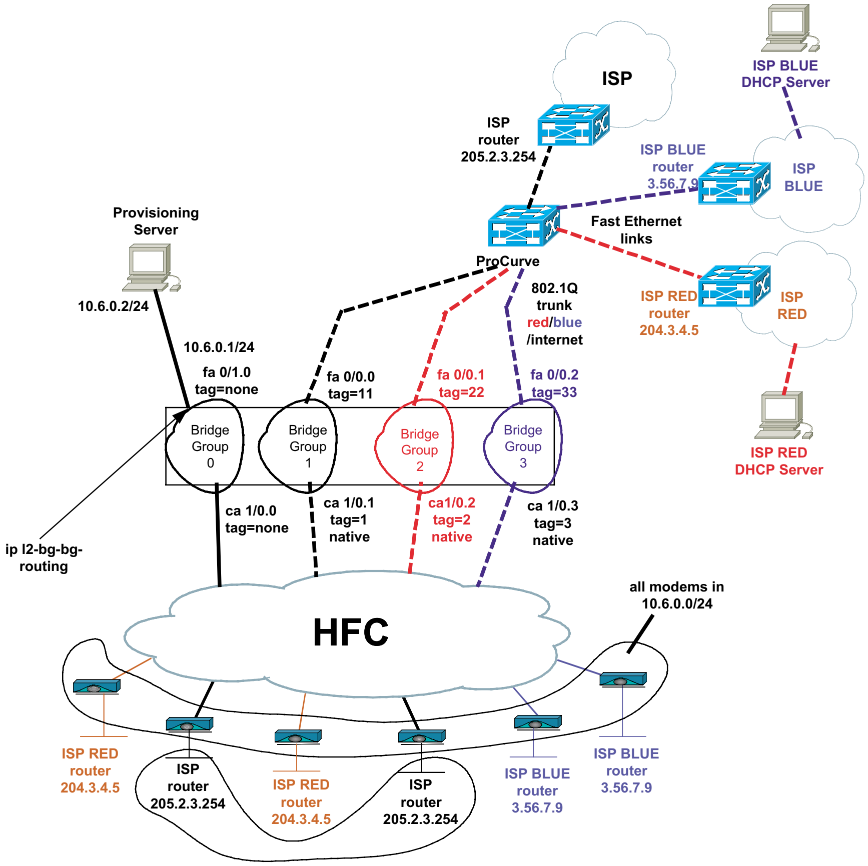
Note: Remember to enable management access.

4 Providing Multiple ISP Access

Open access is an operating concept that allows a subscriber to choose from a number of ISPs. On a practical networking side, open access requires that a subscriber CPE device attached to a cable modem be given a default route that is not associated with any of the cable modem plant. Typically this default route would be the gateway IP address of the chosen ISP's edge router.

Open access support is limited in the C3 to bridging mode only. In IP routing mode, the C3 requires that the CPE device have a default route of the nearest router—in IP routing mode, the nearest router is the C3 cable interface. The C3 as a whole has only has one default route and all CPE traffic would have to use this route thus not allowing an ISP edge router to be selected as the subscriber CPE device default.

The following example shows an open access system implemented with a C3 in bridging mode with three ISPs. Two of the ISPs issue their own IP address; one ISP requires the cable operator to issue CPE IP addresses. In each case, the router option passed to the CPE device is that of the ISP gateway routers and is independent of the cable modem plant



Cable-VPN Implementation

VLANs, combined with the ability to create native VLANs on the cable sub-interfaces may be used to create virtual private networks. In the above example, each subscriber would in effect be provisioned by the cable operator to join one of three virtual private networks, each virtual private network being connected to a single ISP.

Subscribers assigned to an ISP in the above example by the provisioning system can have complete downstream privacy from subscribers assigned to other ISPs, as follows:

- Downstream broadcast privacy
- Downstream unicast privacy
- Upstream unicast privacy
- Upstream broadcast privacy

The following discussion refers to a native VLAN with downstream privacy enabled as a *cable-VPN*.

All physical interfaces may have up to 64 sub-interfaces defined allowing up to 63 native VLANs to be defined per Cadant C3.

Each native VLAN may have downstream privacy enabled.

Example:

```
configure terminal
interface cable 1/0.0
bridge-group 1
encapsulation dot1q 33 native ! create native vlan
encapsulation dot1q 33 encrypted-multicast ! add downstream privacy
exit
```

When this is done, the native VLAN provides downstream privacy for its members and is described following as a cable-VPN.

Cable-VPNs may use IP routing or bridging modes, or both, or may even decode or encode 802.1Q VLANS inside the cable-VPNs as required.

The provisioning systems may assign subscribers to a cable-VPN by the IP address assigned to the modem the subscriber uses or alternatively by the configuration file the modem receives from the provisioning system.

Assignment to a cable-VPN by modem IP address allows legacy provisioning systems to be compatible with the ARRIS Cadant C3 cable-VPN facility. No configuration file modifications are required. This method restricts the number of supported cable-VPNs to 31 (one cable modem sub-interface for every mapped CPE sub-interface) and the DHCP server must support a method to assign a modem an IP address outside the subnet of the giaddr (relay address) in the modem DHCP discover.

Assignment to cable-VPNs by a configuration file allows the full number of 63 cable-VPNs to be implemented but in this case, the DHCP server must support assignment of DHCP options (modem configuration file) to individual modems.

In either case, CPE are mapped to a specific cable sub-interface with native VLAN tagging with the properties of this cable sub-interface defining the properties of the cable-VPN.

- A layer 2 (bridged) cable sub-interface allows all layer 2 protocols inside the cable-VPN.
- When IP routing is active, a layer 3 sub-interface with **ip source-verify subif** specified only allows IP protocols inside the VPN and only source addresses within the subnets associ-

ated with the cable sub-interface (primary subnet and up to 16 secondary subnets per sub-interface).

- A hybrid layer 2 + 3 sub-interface allows both IP and layer 2 protocols.

All cable-VPN sub-interfaces are bridged using bridge groups or IP routed to FastEthernet sub-interfaces.

The C3 FastEthernet sub-interfaces use 802.1Q to propagate the bridged cable-VPN traffic into the operator backplane by maintaining privacy using 802.1Q tagging.

For Open Access purposes, we only consider bridged cable sub-interfaces as discussed above.

Using the Modem IP Address to allocate CPE to a VPN

This example uses the C3 **map-cpes** command.

Modems are issued IP addresses in different subnets. Modems are mapped to cable sub-interfaces by matching the assigned modem IP address to a matching cable sub-interface subnet. Modem cable-sub-interfaces in turn have a **map-cpes** specification that maps all CPE traffic (for CPE attached to these modems) to the cable sub-interface specified by the **map-cpes** command.

Items to note in the following example:

- Select the **no ip routing** mode of operation. This allows the CPE default route or gateway to be specified by the cable operator in the DHCP options given to the CPE and to be different to any IP addressing on the C3. Normally the CPE default route should be directed to the gateway router of the ISP the CPE is to be provisioned to use.

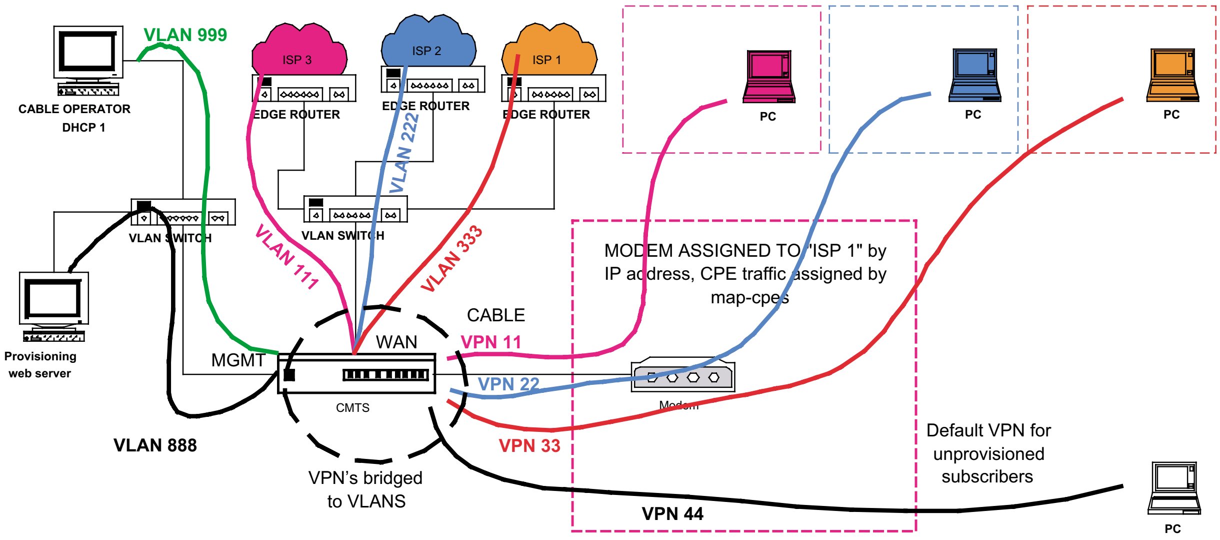
- All CPE traffic is bridged thus layer 2 protocols are supported.

- A default cable-VPN has been created for un-provisioned subscribers. This cable-VPN maps to an Ethernet VLAN directing un-provisioned subscribers to a specific subnet and backbone VLAN allowing access only to the provisioning web server.

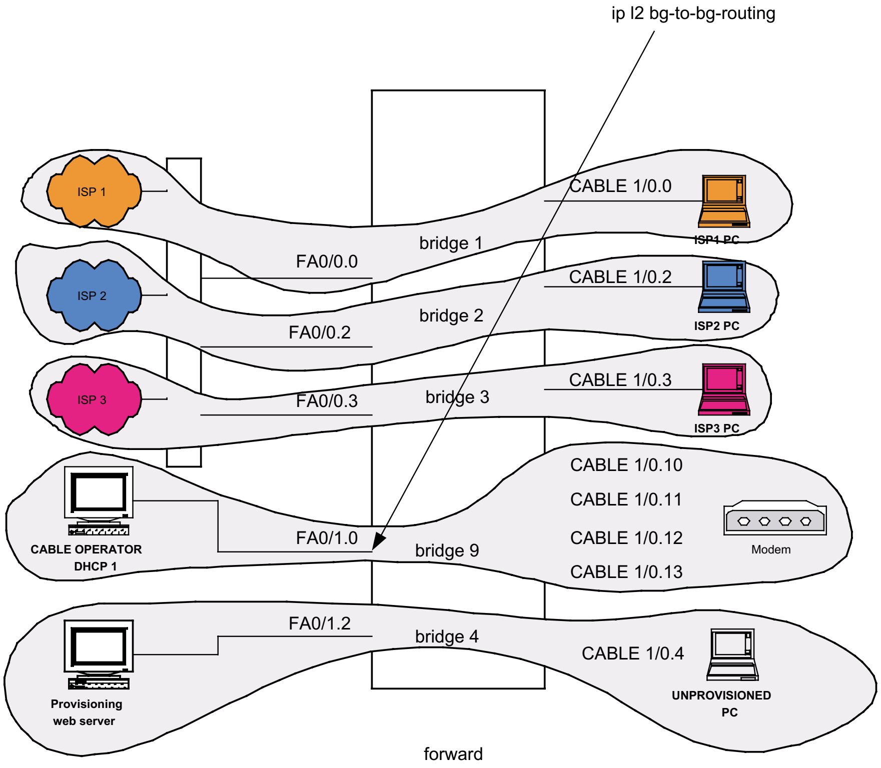
- A default modem cable sub-interface has been created. All modem DHCP discover broadcasts are mapped to this cable sub-interface. This cable sub-interface is a member of bridge group 9. A sub-interface of the MGMT port is configured as a member of this bridge group and has a VLAN tag of 999, the same VLAN tag of the DHCP server.

- Once modems have an IP address, modem traffic is allocated to cable sub-interfaces by modem source IP address match to sub-interface subnet. All modem sub-interface are members of bridge group 9 and are thus connected to the DHCP server using tag 999. These sub-interfaces contain the map-cpes specifications re-directing CPE traffic to other (or the same) cable sub-interfaces and hence cable-VPNs.

The following shows the network diagram for this example.



The following shows how the C3 bridges data flowing through the above network.



Configuration

```
! run the following as a script on a factory default C3 configuration
!
conf t
!
! remove the factory default assignments
!
! remove bridges 0 and 1 so no sub-interfaces are attached
no bridge 0
no bridge 1
int ca 1/0
!
! remove any previous ip addresses from the cable interface
no ip address 10.99.99.253 255.255.255.0
exit
! remove the cable 1/0.1 subinterface
! as factory defined but not going to be used
no int ca 1/0.1
!
no ip routing
!
! set default subinterface for cm and cpe taffic
!  before  cm has an IP address
default-cm-subinterface cable 1/0.10
```

```
! catch any unknown CPE and direct to
! the provisioning web server
default-cpe-subinterface cable 1/0.4
!
! Define the bridges we will use
! for ISP1 traffic
bridge 1
! for ISP2 traffic
bridge 2
! for ISP3 traffic
bridge 3
! for provisioning server traffic
bridge 4
! bridge 9 used for cm dhcp discover
! and management access to CMTS
! all cm will have access to this bridge group no
! matter what ip address they end up with
bridge 9
!
int fa 0/0.0
description ISP1
! no ip address
bridge-group 1
encapsulation dot1q 111
no ip l2-bg-to-bg-routing
exit
!
int fa 0/0.2
description ISP2
! no ip address
bridge-group 2
encapsulation dot1q 222
no ip l2-bg-to-bg-routing
exit
!
int fa 0/0.3
description ISP3
! no ip address
bridge-group 3
encapsulation dot1q 333
no ip l2-bg-to-bg-routing
exit
!
interface fa 0/1.0
description Management
ip address 10.99.99.2 255.255.255.0
! NOTE: CMTS management can only occur from this VLAN
encapsulation dot1q 999
management-access
bridge-group 9
```

```
ip l2-bg-to-bg-routing
! ip address should be in subnet of DHCP server
! this is also the CMTS management address
!
! DHCP server should have static routes added
! for each CPE subnet with this address as the gateway
! e.g.
!     route add 10.1.0.0 mask 255.255.255.0 10.99.99.2
!     route add 10.2.0.0 mask 255.255.255.0 10.99.99.2
!     route add 10.3.0.0 mask 255.255.255.0 10.99.99.2
! so that CPE DHCP ofer and ack can be routed back to
! the appropriate bridge group and hence CPE device
! Note: dhcp relay must be active in all CPE bridge
! groups for this to happen and only DHCP will be routed
exit
!
interface fa 0/1.2
description Provisioning
! ip address should be a subnet
! of provisioning web server
ip address 10.88.88.2 255.255.255.0
encapsulation dot1q 888
no management-access
bridge-group 4
no ip l2-bg-to-bg-routing
exit
!
interface cable 1/0.0
description ISP1_CPE
ip address 10.1.0.1 255.255.0.0
! Note: up to 16 secondary IP addresses can be added
! for non contigous ISP subnets
no management-access
! set up dhcp relay for CPE devices
! must have dhcp relay active in each bridge group
! for dhcp to be forwarded across the bridge groups
! to the dhcp server in bridge-group 9
ip dhcp relay
cable helper address 10.99.99.1
no ip dhcp relay information option
! native tagging required for internal processing
encapsulation dot1q 1 native
! turn on downstream broadcast privacy
encapsulation dot1q 1 encrypted-multicast
bridge-group 1
no ip l2-bg-to-bg-routing
exit
!
interface cable 1/0.2
description ISP2_CPE
```

```
ip address 10.2.0.1 255.255.0.0
no management-access
ip dhcp relay
cable helper address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 2 native
! turn on downstream broadcast privacy
encapsulation dot1q 2 encrypted-multicast
bridge-group 2
no ip l2-bg-to-bg-routing
exit
!
interface cable 1/0.3
description ISP3_CPE
ip address 10.3.0.1 255.255.0.0
no management-access
ip dhcp relay
cable helper address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 3 native
! turn on downstream broadcast privacy
encapsulation dot1q 3 encrypted-multicast
bridge-group 3
no ip l2-bg-to-bg-routing
exit
!
interface cable 1/0.4
description UNPROVISIONED_CPE
! ip address should be in the subnet of the
! provisioning server
ip address 10.4.0.1 255.255.0.0
no management-access
ip dhcp relay
cable helper address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 4 native
! turn on downstream broadcast privacy
ecnapsulation dot1q 4 encrypted-multicast
bridge-group 4
no ip l2-bg-to-bg-routing
exit
!
interface cable 1/0.10
description modem_default
! default for cm devices before they have IP address
ip address 10.77.77.1 255.255.255.0
no management-access
encapsulation dot1q 10 native
bridge-group 9
ip address 10.77.77.1 255.255.255.0
```

```
no management-access
! set up dhcp relay for cm
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
! map attached CPE to the provisioning server
! if a cm is stil lusing this subinterface
! then cm has not been provisioned yet
map-cpes cable 1/0.4
!
exit
!
interface cable 1/0.11
description modem_isp1
! for cm devices for ISP 1 once cm has IP address
ip address 10.11.0.1 255.255.0.0
encapsulation dot1q 11 native
bridge-group 9
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no management-access
! map all cpe traffic
map-cpes cable 1/0.1
exit
!
interface cable 1/0.12
description modem_isp2
! for cm devices for ISP 2 once cm has IP address
ip address 10.12.0.1 255.255.0.0
encapsulation dot1q 12 native
bridge-group 9
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no management-access
map-cpes cable 1/0.2
exit
!
interface cable 1/0.13
description modem_isp3
! for cm devices for ISP 3 once cm has IP address
ip address 10.13.0.1 255.255.0.0
encapsulation dot1q 13 native
bridge-group 9
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no management-access
```

```
map-cpes cable 1/0.3
exit
!
interface cable 1/0.0
! Get rf running
! not no rf configuration here so check the factory
! defaults are ok
no cable upstream 0 shutdown
no cable upstream 1 shutdown
no shutdown
no management-access
! no ip address as sub-interface is not used
exit
!
exit
```

Using a Modem Configuration File to Allocate CPEs to a VPN

This example uses the Cadant C3 Vendor Specific Encoding in the modem configuration files to map CPE attached to modems to specific cable sub-interfaces and hence to specific cable-VPNs and backbone 802.1Q VLANs.

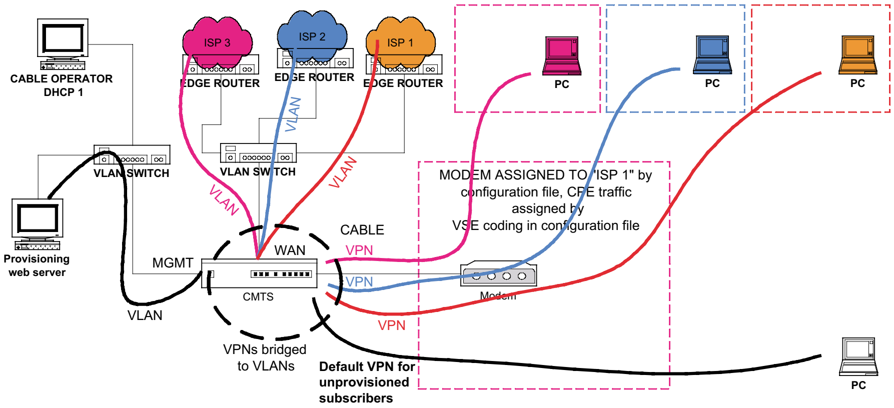
The following example:

- Uses fewer (one only) cable sub-interfaces for modems than the map-cpes method
- Uses VSE encoding to map CPE traffic to cable sub-interfaces with native VLAN specifications (cable-VPN) and hence to bridge-groups and hence to Ethernet sub-interfaces and hence to Ethernet backbone 802.1Q VLANS.

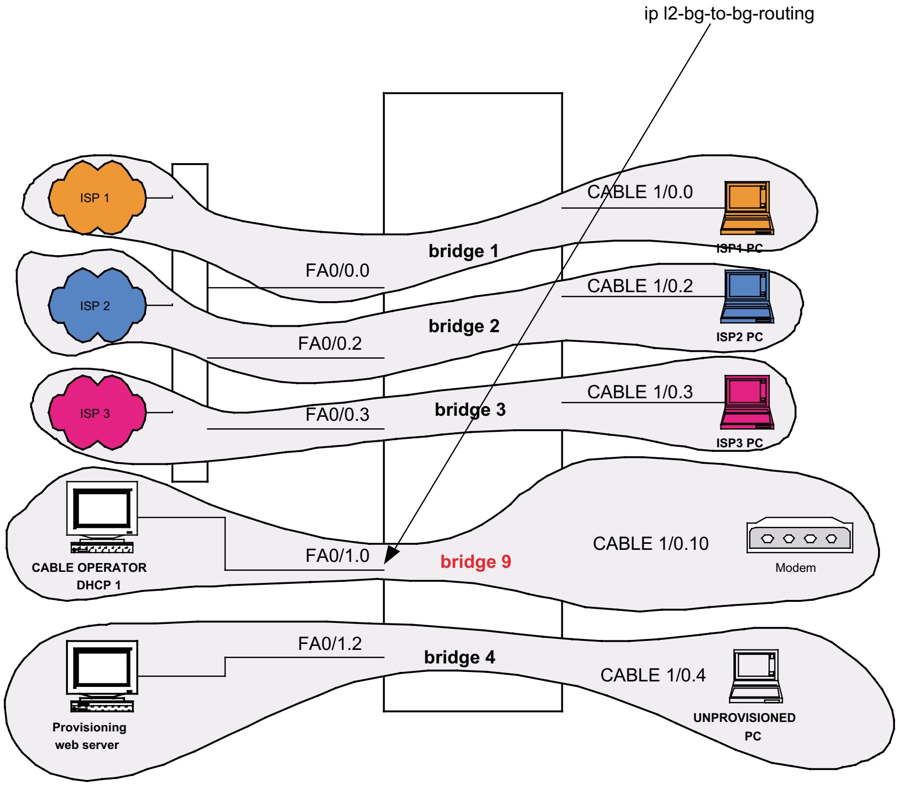
Items to note in the following example:

- A default cable-VPN has been created for un-provisioned subscribers. Modems given a configuration file with a VSE encoding of 44 will force attached CPE devices to the backbone 802.1Q VLAN with a tag of 888. This Ethernet VLAN connects to the provisioning web server.
- A default modem cable sub-interface has been created. All modem traffic before an IP address is allocated to the modem is mapped to this cable sub-interface. This cable sub-interface is a member of bridge group 9. A sub-interface of the MGMT port is configured as a member of this bridge group and has a VLAN tag of 999. As there are no sub-interfaces defined with matching subnets to that allocated for modems, all modem traffic will remain mapped to this interface.

The following shows the diagram of the network used for this example:



The following shows how the C3 bridges data in the example network:



Configuration

As can be seen following the level of configuration required is lower than the map-cpes method.

Notable differences are:

- All modems are now contained in the one IP subnet. This requires that the DHCP server must support the specification of DHCP options per reserved address.

- The encapsulation “native” commands in cable sub-interfaces 0.1 through 1/0.3 must match the VSE tagging. If no match is found, the CPE traffic will be mapped to the default cable 1/0.4 sub-interface and be bridged to the provisioning web server.
- Again option 82 processing is turned off but may be turned on again if an option 82 aware DHCP server is to be used.

```
! run the following as a script on a factory default C3 configuration
!
conf t
! remove bridges 0 and 1 so no sub-interfaces are attached
no bridge 0
no bridge 1
!
int ca 1/0
! remove any previous IP addresses from the cable interface
no ip address 10.99.99.253 255.255.255.0
exit
! remove the cable 1/0.1 subinterface -- not used
no int ca 1/0.1
!
no ip routing
!
! set default subinterface for cm taffic before
! cm has an IP address
default cm subinterface cable 1/0.10
default cpe subinterface cable 1/0.4
!
! Define the bridges we will use for CPE trafic
bridge 1
bridge 2
bridge 3
bridge 4
bridge 9
!
int fa 0/0.0
! description ISP1_WAN
encapsulation dot1q 111
bridge-group 1
exit
!
int fa 0/0.2
! description ISP2_WAN
encapsulation dot1q 222
bridge-group 2
exit
!

int fa 0/0.3
```

```
! description ISP3_WAN
encapsulation dot1q 333
bridge-group 3
exit
!
interface fa 0/1.0
! description MANAGEMENT
! ip address should be in subnet of DHCP server
ip address 10.99.99.2 255.255.255.0
management-access
encapsulation dot1q 999
bridge-group 9
ip l2-bg-to-bg-routing
exit
!
interface fa 0/1.2
! description PROVISIONING_SERVER
! ip address should be subnet of provisioning web server
ip address 10.88.88.2 255.255.255.0
encapsulation dot1q 888
no management-access
bridge-group 4
exit
!

interface cable 1/0.0
! description ISP1_CPE
ip address 10.1.0.1 255.255.0.0
no management-access
! set up dhcp relay for CPE devices
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
! VSE tagging
! all cm with VSE tag of 11 will cause all attached
! CPE to be mapped to this interface
encapsulation dot1q 11 native
! turn on VPN
encapsulation dot1q 11 encrypted-multicast
bridge-group 1
exit
!
interface cable 1/0.2
! description ISP2_CPE
! for CPE devices for ISP2
ip address 10.2.0.1 255.255.0.0
no management-access
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 22 native
encapsulation dot1q 22 encrypted-multicast
bridge-group 2
exit
!
interface cable 1/0.3
! description ISP3_CPE
! for CPE devices for ISP3
ip address 10.3.0.1 255.255.0.0
no management-access
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 33 native
encapsulation dot1q 33 encrypted-multicast
bridge-group 3
exit
!
interface cable 1/0.4
! description UNPROVISIONED_CPE
! for CPE devices for unprovisioned subscribers
ip address 10.4.0.1 255.255.0.0
no management-access
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 44 native
encapsulation dot1q 44 encrypted-multicast
bridge-group 4
exit
!
interface cable 1/0.10
! default for cm devices
! all cm will remain on this interface
bridge-group 9
ip address 10.77.77.1 255.255.255.0
no management-access
! set up dhcp relay for cm
! note: dhcp relay is not really required as DHCP bcast
! would be bridged to the DHCP server network
! via bridge group 9
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
exit
```

```
!
interface cable 1/0
! Get rf running
! not no rf configuration here so please check the factory
! defaults are ok
no cable upstream 0 shutdown
no cable upstream 1 shutdown
no shutdown
no management-access
! no ip address as sub-interface is not used
exit
!
exit
!----------- end script ----------------
```

An extension—no Ethernet VLANs used

Where the Ethernet backbone does not have VLAN support, Open Access is still possible.

A reminder of some rules to begin with—rules that drive the following configuration.

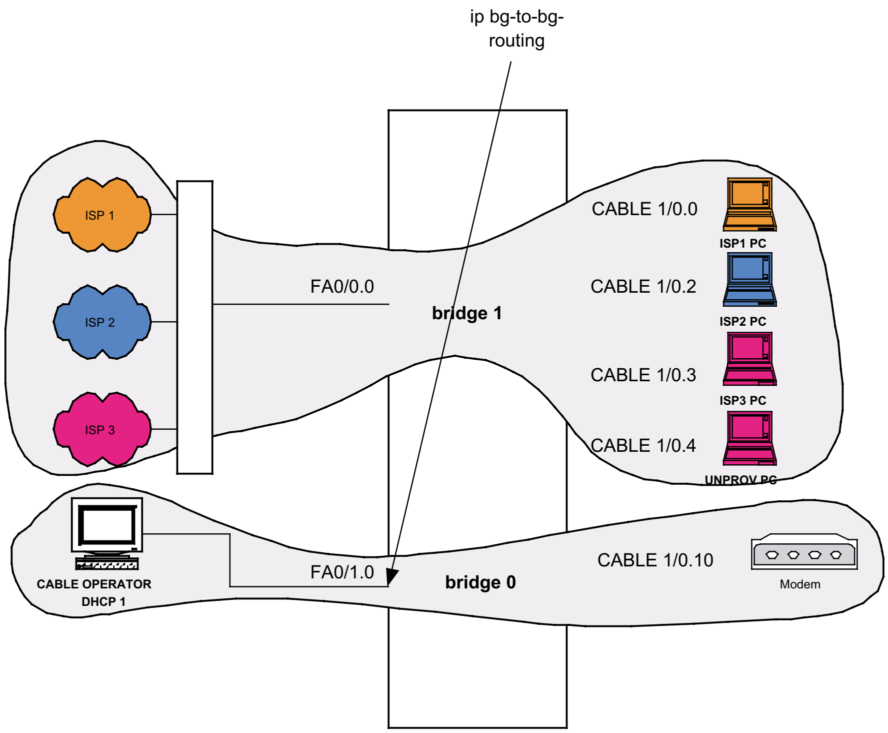
- One sub-interface on a physical interface may be untagged.
- There is a maximum of 10 sub-interfaces per any single bridge-group.
- Up to 64 sub-interfaces may be defined for each physical interface.
- Up to 64 bridge-groups may be defined.
- DHCP relay operates across bridge groups but must be turned on in the bridge groups where it is required. If turned on, the DHCP relay supporting sub-interface must have at least one IP address specification—even if bridging all other traffic.

With reference to this specific configuration example:

- There is a maximum of 10 sub-interfaces per any single bridge group.
- CPE cable sub-interfaces are created and are made members of bridge group 1.
- For bridge group 1 to access the Ethernet backbone, an Ethernet sub-interface must also be a member of this bridge group.
- All Cable CPE sub-interfaces are added to bridge group 1 that now has untagged access to the Ethernet backbone.

- A maximum of 9 CPE sub-interfaces may be supported in this manner. Thus a maximum of 9 cable-VPNs may be supported with this configuration.
- If DHCP relay is required, **ip dhcp relay** must be turned on and for IP DHCP relay to function, the CPE sub-interface must have at least one IP address specification. If the CPE are to receive IP address from the operator DHCP server, **l2 bg-to-bg-routing** must be turned on to allow forwarded DHCP to pass across the boundary of bridge group 1 to bridge group 0.

The following shows how the C3 bridges data in this configuration:



Configuration

```
conf t
! remove bridges 0 and 1 so no sub-interfaces are attached
no bridge 0
no bridge 1
!
int ca 1/0
! remove any previous ip addresses from the
! cable interface
no ip address 10.99.99.253 255.255.255.0
exit
! remove the cable 1/0.1 subinterface
! not used
no int ca 1/0.1
```

```
!
no ip routing
!
! set default subinterface
default cm subinterface cable 1/0.10
default cpe subinterface cable 1/0.4
!
! Define the bridges we will use
bridge 0
bridge 1
!
int fa 0/0.0
! description ISP_WAN
bridge-group 1
exit
!
interface fa 0/1.0
! description MANAGEMENT
bridge-group 0
ip l2-bg-to-bg-routing
! ip address should be in subnet of DHCP server
ip address 10.99.99.2 255.255.255.0
management-access
exit
!
interface cable 1/0.0
! Get basic rf running
no cable upstream 0 shutdown
no shutdown
no management-access
! description ISP1_CPE
! for CPE devices for ISP1
ip address 10.1.0.1 255.255.0.0
no management-access
! set up dhcp relay for CPE devices
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
! all cm with VSE tag of 11 will cause all attached
! CPE to be mapped to this interface
encapsulation dot1q 11 native
! add to bridge group to get bridged eth access
bridge-group 1
exit
!
interface cable 1/0.2
! description ISP2_CPE
! for CPE devices for ISP2
ip address 10.2.0.1 255.255.0.0
```

```
no management-access
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 22 native
bridge-group 1
exit
!
interface cable 1/0.3
! description ISP3_CPE
! for CPE devices for ISP3
ip address 10.3.0.1 255.255.0.0
no management-access
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 33 native
bridge-group 1
exit

!
interface cable 1/0.4
! description UNPROVISIONED_CPE
! for CPE devices for unprovisioned subscribers
ip address 10.4.0.1 255.255.0.0
no management-access
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
encapsulation dot1q 44 native
bridge-group 1
exit
!
interface cable 1/0.10
! default for cm devices
! all cm will remain on this interface
ip address 10.77.77.1 255.255.255.0
no management-access
! set up dhcp relay for cm
ip dhcp relay
cable dhcp-giaddr primary
cable helper-address 10.99.99.1
no ip dhcp relay information option
exit
!
exit
```

5 IP Routing

This chapter describes Layer 3 (routing) operation of the Cadant C3 CMTS.

See Appendix B for a routing configuration example.

Routing Concepts

A quote from RFC 2453: “Routing is the task of finding a path from a sender to a desired destination.”

IP packets contain a source and destination IP address. But an IP packet is transported using lower layer protocols and these link-layer protocols require a destination hardware (MAC) address to forward the packet.

When the destination IP address is on a network directly connected to the C3, the C3 can send a broadcast message (ARP) to the subnet asking “whoever owns this IP address, please give me your hardware address.”

Default Route

When the destination subnet is not known to the C3, the C3 does not know what to do with the packet unless a route is present. If no other route is present, the **ip route 0.0.0.0 0.0.0.0 a.b.c.d** command can be used to tell the C3 to pass the packet to this gateway of last resort—IP address **a.b.c.d** in this example.

This default gateway also may not know how to route the packet. In this case, the gateway may return the ICMP “host unreachable” or “destination unreachable” message if the gateway routing policies allow any such response.

The gateway device is normally a router, and the unknown subnet may be on the other side of this router. This other device would also normally have knowledge of the network topology far beyond its own interfaces. Such knowledge could be propagated between such routing devices by RIP (Routing Information Protocol). There are many other routing protocols, but the C3 currently supports only RIP.

Static Routing

Static routing involves manually configuring routes to certain IP hosts, using the **ip route** command. If you are not using learned (dynamic) routing, you must configure a static route to the default gateway device using the **ip route** command. Use the **ip route** command to provide a route to a destination network or to a destination host. The **ip route 0.0.0.0 0.0.0.0 a.b.c.d** command is a special form of this command used to set a default route as discussed above.

Different gateways may be given for the same route with different administrative distances—the C3 uses the route with the lowest administrative distance until the route fails, then uses the next higher administrative distance, and so on. Up to 6 static routes may be configured in this manner. The route to a connected subnet (subnet of a sub-interface) always has an administrative distance of **0** and thus takes precendence over any static route.

In case of two static routes to the same prefix with equal administrative distance, the C3 uses the first provisioned route. If that route fails, then the C3 uses the next route. After rebooting, the C3 uses the first static route defined in the startup-configuration file. An example of this is shown in “Routing Priority” on page 5-3—refer to the 6 static routes (*) and (**) for network 15.0.0.0/24.

Static routing is supported in all C3 operating modes.

Dynamic Routing

Learned routing, or *dynamic routing*, means that the C3 learns routes to various destinations from messages sent by other routers on the network. In this version of C3 operating software, the C3 supports RIPv1 and RIPv2 (RFC1812) for learning routes.

About RIP

RIP (Routing Information Protocol) is a *de facto* standard for exchanging routing information between routers and gateway devices.

To enable RIP in the C3, see “Routing Command Overview” on page 5-6.

The benefits of enabling RIP in the C3 are:

- You no longer need to specify a default gateway to let the C3 find distant destinations; the C3 learns about the network topology around it using RIP.
- Other devices on the Internet backbone use information from the C3 (through RIP) to learn how to contact cable interface subnets behind the C3.

RIP routing is an extra-cost option. Contact your ARRIS representative to obtain a license key.

Routing Priority

Use the **show ip route** command to display routing priority. In the following example, comments have been added using “<<<<<” to add some further clarification to the output:

```
C3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - ICMP, B - BGP
       E - EGP, G - GGP, O - OSPF, ES - ES-IS, IS - IS-IS
       * - candidate default, > - primary route

Gateway of last resort is 10.250.96.1 to network 0.0.0.0


S*   0.0.0.0/0 [1/0] via 10.250.96.1, FastEthernet 0/1.0
     4.0.0.0/24 is subnetted, 1 subnet
R          4.4.4.0 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
<<<<< rip learned - default AD=120
     5.0.0.0/24 is subnetted, 1 subnets
S>      5.5.5.0 [130/0] via 10.250.96.7, FastEthernet 0/1.0
<<<< primary static with AD changed to 130
S            [130/0] via 10.250.96.8, FastEthernet 0/1.0
<<<< backup static
     7.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
R       7.0.0.0/24 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
R       7.0.0.0/8 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
R       7.7.0.0/16 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
     10.0.0.0/24 is subnetted, 4 subnets
C        10.7.8.0 is directly connected, Cable 1/0.9
<<<< directly connected to c3 (configured on sub-int AD=0)
C          10.250.96.0 is directly connected, FastEthernet 0/1.0
C          10.250.99.0 is directly connected, FastEthernet 0/0.0
C          10.250.103.0 is directly connected, bridge-group #0
     15.0.0.0/24 is subnetted, 1 subnets
S>       15.5.5.0 [1/0] via 10.7.8.10, Cable 1/0.9
<<< static with default AD=1 (*)
S                  [1/0] via 10.7.8.11, Cable 1/0.3
<<<< backup static, AD=1, second in config file (**)
S                   [1/0] via 10.7.8.110, Cable 1/0.3
<<<< backup static, AD=1, 3 in config file (**)
S                   [1/0] via 10.71.8.11, Cable 1/0.30
<<<< backup static, AD=1, 4 in config file (**)
S                    [1/0] via 10.72.8.11,  FastEthernet 0/0.5
<<<< backup static, AD=1, 5 in config file (**)
S                   [1/0] via 100.78.8.11, Cable 1/0.23
<<<< backup static, AD=1, 6 in config file (**)
     79.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R       79.79.79.0/24 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
R       79.79.79.101/32 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
```

Note the two numbers in brackets shown for each defined route:

- The first number is the administrative distance of the route. *Connected routes* (meaning a C3 sub-interface has an IP address within this subnet) have an administrative distance of 0; static routes have a default distance of 1. Routes learned through RIP have a default distance of 120.

- The second number is the route metric, which is significant only for RIP routes.

When there are several paths to a destination IP address, the C3 uses the following scheme to determine routing priority:

- Connected routes always have priority over static routes.

- Static routes always have priority over dynamic routes.

- The most specific route—that is, the route with the longest prefix (smallest subnet size) has the highest priority.

- Given equally specific static routes, the C3 chooses the path with the lowest administrative distance.

- Given both equally specific static routes with equal administrative distances, the C3 uses the first provisioned route. If that route fails, then C3 uses the next route. Up to 6 routes are supported in this manner.

  After a reboot, the C3 uses the first of these static routes in the startup-configuration file.

- Given both equally specific dynamic routes and equal administrative distances, the C3 chooses the route with the lowest metric number.

- Given both equally specific dynamic routes with equal administrative distances and equal metrics, per RFC2453, the C3 uses the first dynamic route until it fails (failure detected after 90 seconds using default RIP timers—180/2 seconds).

Routing Authentication

Dynamic routing protocols such as RIP build a network topology using updates received from other routers. On a cable data network, a subscriber could potentially connect a router to a cable modem then advertise spoofed routes to other networks.

Authentication prevents malicious subscribers (or other entities) from polluting the C3's network topology with bogus information. The C3 uses a key chain that supports automatically changing keys over time. The authentication system is similar to that supported by Cisco routers.

Key Chains

Key chains consist of one or more keys. Each key in a key chain is a 16-character string or an MD5 key, and can be sent to other routers or accepted from other routers; the default is to both send and receive keys. In addition, each key can have a send or accept lifetime, allowing for a rotation of valid keys over time.

See “key chain” on page 6-90 for more details about configuring key chains.

Enabling RIP Authentication

Use the ip rip authentication command on a sub-interface to specify a key chain, text password, or MD5 password to accept from other routers in the network.

See “ip rip authentication” on page 6-115 for details about the command.

Routing Command Overview

The only routing commands required are:

```
C3(config)# ip routing
C3(config)# router rip
C3(config-router)# network subnet wildcard
```

Where *subnet* is a standard subnet address, and *wildcard* is an inverted mask (for example, if the mask is **255.255.255.0**, the wildcard is **0.0.0.255**).

Tip: to enable RIP on all sub-interfaces, use the command **network 0.0.0.0 255.255.255.255**

Other routing parameters have reasonable defaults for most network configurations; for example, RIP version 2 is run by default.

Note: When configuring routing from a telnet session, you also need to specify a default route using the **ip route** command before starting IP routing. This allows the C3 to continue the telnet session so you can enter other routing commands while the C3 learns the route back to your system.

RIP-related routing commands fall into two categories:

- general: described in “Router Configuration Mode” on page 6-144.
- sub-interface specific: described in “Common Interface Sub-commands” on page 6-111.

6 Command Line Interface Reference

The Cadant C3 command line interface (CLI) is intended to follow the familiar syntax of many other communications products and to provide ease of use for administrators.

CLI Modes

The user interface operates in the following modes:

- User mode—This is the initially active mode when a user logs into the CLI. The user is limited to harmless commands, such as changing the terminal setting, pinging a host, or displaying certain configuration information.
- Privileged mode—Type **enable** and enter a valid password in order to enter privileged mode. In privileged mode, all the commands of user mode are available, along with extra commands for debugging, file manipulation, diagnostics, and more detailed configuration display.
- Configure mode—Type **configure** while in privileged mode to enter Configure mode. In configure mode, the commands available relate to general system configuration and are not specific to any particular interface. Cable modem commands are also available in configure mode.
- Configure interface sub-modes—To configure a particular interface, enter a configuration sub-mode by typing the appropriate command from Configure mode. The currently available interfaces are terminal, fastethernet, and cable.
- Router configuration mode—To configure routing parameters, routing configuration mode must be entered.

Command Completion and Parameter Prompting

Press the **Tab** key to complete a partially-typed command. If what you type previous to the **Tab** could be completed in two different ways (for example, **co** could be completed as **configure** or **copy**), the C3 console beeps and does not attempt to complete the command.

Example:

```
# con<tab>
# configure
```

The ? (question mark) key has two purposes:

- When added to the end of a partially-typed command, the C3 lists commands that start with the current fragment.
- When separated from the command by one or more spaces, the C3 lists valid parameters or values that can follow the command.

Example:

```
(config)#lo?
logging  login
(config)#logging ?
buffered          - Enable local logging of events in a circular buffer
on                - Enable all logging
severity          - Enable/disable logging for a particular severity
syslog            - Enable syslog logging for events
thresh            - Configure thresholds
trap              - Enable traps
trap-control      - Configure DOCSIS trap control

(config)#logging
```

Input Editing

Use the following keystrokes to edit a command before entering it.

Character sequence	**Common Name**	**Action**
<CR>	Carriage Return	Passes completed line to parser
<NL>	Newline	Passes completed line to parser
<DEL>	Delete	Backspace one character and delete
?	Question Mark	Provides help information
^A	Control-A	Position cursor to start of line

Character sequence	Common Name	Action
^B	Control-B	Position cursor left one character
^C	Control-C	Telnet session: Clears input and resets line buffer. Serial console: Opens low-level console (prompting for password).
^D	Control-D	Delete current character
^E	Control-E	Position cursor to end of line
^F	Control-F	Position cursor right one character
^H	Control-H	Backspace one character and delete
^I	Tab	Complete current keyword
^K	Control-K	Delete to end of line
^L	Control-L	Redraw line
^N	Control-N	Move down one line in command history
^P	Control-P	Telnet session: Move up one line in command history.
^R	Control-R	Redraw line
^U	Control-U	Clears input and resets line buffer.
^X	Control-X	Clears input and resets line buffer.
^Z	Control-Z	Pass control to user session exit function
<ESC>[A	Up Arrow	Move up one line in command history
<ESC>[B	Down Arrow	Move down one line in command history
<ESC>[C	Right Arrow	Position cursor right one character
<ESC>[D	Left Arrow	Position cursor left one character
<SP>	Space	Separates keywords
"	Quote	Surrounds a single token
^W	Control-W	Delete the last word before the cursor on the command line

Output Filtering

The C3 provides output filtering commands. You can use them to reduce the amount of output sent to the screen by certain commands.

You specify output filtering by appending a vertical bar character to the end of a command, followed by the filtering command and its arguments. The output filtering commands are **begin**, **include**, and **exclude**. The **?** (help) command prints a brief summary of the commands:

```
C3#show run | ?
begin    Begin with the line that matches
include  Include lines that match
exclude  Exclude lines that match
```

Filtering Previous Lines

Use the **begin** command to suppress output until an output line matches the specified string:

```
C3#show run | begin "interface Cable"
interface Cable 1/0
 cable insertion-interval automatic
 cable sync-interval 10
 cable ucd-interval 2000
! cable max-sids 8192
 cable max-ranging-attempts 16
 cable map-advance static
 cable downstream annex B
```

etc…

Including Matching Lines

Use the **include** command to display only output lines matching the specified string:

```
C3#show access-lists interface matches | include “Outgoing”
FastEthernet 0/0        Outgoing              78        None Set  N/A
FastEthernet 0/1        Outgoing         Not Set        None Set  N/A
Cable 1/0               Outgoing             171               1  0
Cable 1/0               Outgoing             171               2  0
Cable 1/0               Outgoing             171               3  0
Cable 1/0               Outgoing             171               4  0
Cable 1/0               Outgoing             171               5  0
Cable 1/0               Outgoing             171               6  1529
Cable 1/0               Outgoing             171               7  1482
Cable 1/0               Outgoing             171               8  186184
```

Excluding Matching Lines

Use the **exclude** command to suppress output lines matching the specified string:

```
C3#show access-lists interface matches | exclude “FastEthernet”
Interface               Direction         Acl ID        Entry No.Matches
Cable 1/0               Outgoing             171               1  0
Cable 1/0               Outgoing             171               2  0
Cable 1/0               Outgoing             171               3  0
Cable 1/0               Outgoing             171               4  0
Cable 1/0               Outgoing             171               5  0
Cable 1/0               Outgoing             171               6  1529
Cable 1/0               Outgoing             171               7  1482
Cable 1/0               Outgoing             171               8  186184
Cable 1/0               Inbound             2601        None Set  N/A
```

User Mode Commands

User mode is in effect when you log into the CMTS. Commands in this mode are limited to inquiry commands. The prompt in user mode is the hostname followed by a greater than sign (e.g., hostname>).

The following is a summary of user mode commands:

```
C3>?
enable             -
exit               - Exit Mode / CLI
help               - Display help about help system
llc-ping           - Ping a specific MAC address using 802.2 LLC TEST frames
logout             - Exit the CLI
ping               - Ping a specific ip address
show               - Show system info
systat             - Display users logged into CLI
terminal           - Change terminal settings
*scm               - Alias: "show cable modem"
C3>
```

enable

Enters privileged mode.

See “Privileged Mode Commands” on page 6-16 for more details. You need to use the enable password to enter privileged mode.

exit

In user mode, terminates the console session.

help

Provides a list of the available commands for the current user mode.

llc-ping

Syntax: **llc-ping {macaddr }[continuous | n]<inter-ping-interval-in-seconds>**

Sends a series of MAC-level echo requests to the specified modem MAC address, and reports whether the CMTS received an echo response for each packet. This command runs until you press a key or until the C3 has sent the specified number of pings.

> ***Note:*** Not all cable modems or MTAs respond to **llc-ping**.

```
C3#>llc-ping 1111.1111.1111 continuous 5
C3#>llc-ping 1111.1111.1111. 6 7
```

logout

Closes the connection to the CMTS regardless of operating mode.

ping

Syntax: **ping {ipaddr}**

Sends a series of 5 ICMP echo requests to the specified IP address, and reports whether the CMTS received an echo response for each packet.

show

Displays information about the system. The following options are available:

```
C3>show ?
aliases            - Show aliases
arp                - ARP table
bootvar            - Show boot parameters
calendar           - Show Date and Time
clock              - Show Date and Time
context            - Context info about recent crashes
exception          - Show information from the autopsy file
hardware           - Hardware information
history            - Command History
ip                 - IP related info
ipc                - IPC info
key                - Key Information
memory             - System memory
ntp                - NTP Servers
snmp               - SNMP counters
terminal           - Terminal info
tftp-server        -
users              - Users logged into CLI
version            - Version information

C3>
```

show aliases

Displays any defined aliases for commands.

See also: “alias” on page 6-67.

```
C3>show alias

=Alias=             =Command string=

scm                 show cable modem
```

show arp

Equivalent to the **show ip arp** command without arguments.

Example:

```
C3>show arp
Prot Address           Age(min) Hardware Addr  Vlan Type Interface
```

```
IP   10.1.176.193    15       0001.5c20.4328   -  ARPA B#0-FastEthernet 0/0.0
IP   10.1.176.254    0        00e0.168b.fc89   -  ARPA B#0-FastEthernet 0/0.0
C3#
```

show bootvar

Displays boot variables.

```
C3>show bootvar

Boot Image Device: Compact Flash - C:/3.0.1.27.bin
Boot Config file Device: current flashdisk file
C3>
```

See also: “boot system flash” on page 6-67 (privilege mode required).

show calendar

Displays the date and time from the internal real time clock. The internal clock has a battery backup and operates whether or not the C3 is powered down.

```
C3>show calendar
20:13:38 GMT Tue Aug 27 2002
20:13:38 UTC Tue Aug 27 2002
C3>
```

See also: “clock timezone” on page 6-84.

show clock

Displays the date and time from the system clock. The C3 synchronizes the system clock with the calendar at boot time.

```
C3>show clock
15:54:27.481 GMT Tue Jul 15 2003
15:54:27.481 UTC Tue Jul 15 2003
C3>
```

See also: “clock timezone” on page 6-84.

show clock timezone

Displays the current time zone and its offset from GMT.

```
C3>show clock timezone
Local time zone is GMT (0:00 from UTC)
C3>
```

show context

Displays recent startup and shutdown history.

Example:

```
C3>show context
Shutdown: Date Tue 08-Jul-2003: time 02:27:54
Bootup  : Date Tue 08-Jul-2003: time 02:29:55
Bootup  : Date Wed 09-Jul-2003: time 01:38:21
Shutdown: Date Wed 09-Jul-2003: time 03:00:26
Bootup  : Date Wed 09-Jul-2003: time 03:01:16
```

show exception

Identical to **show context**.

show hardware

Displays a list of hardware installed in the CMTS with revision information and serial numbers where appropriate.

Example:

```
C3>show hardware
Arris C3 CMTS - Serial # 312
Component   Serial #     HW Rev       SW Rev
WAN/CPU     000312       unavailable  N/A
Cable       N/A          A            N/A
Upconverter N/A          6            N/A
Extender    N/A          2            7
FPGA S/W    N/A          N/A          5

Processor Module BCM1250
CPU       : 1250 A8/A10
Nb core   : 2
L2 Cache  : OK
Wafer ID  :   0x2C6C4019  [Lot 2843, Wafer 2]
Manuf Test: Bin A [2CPU_FI_FD_F2 (OK)]
Cpu speed : 600 Mhz
SysCfg    : 000000000CDB0600 [PLL_DIV: 12, IOB0_DIV: CPUCLK/4,
IOB1_DIV: CPUCLK/3]

Downstream Module BCM3212(B1)
        Description: DS 1 - Cadant C3 CMTS - BCM3034 Rev A1
Upstream modules
        Description: US CH 1 - Cadant C3 CMTS - BCM3138 Rev A2
        Description: US CH 2 - Cadant C3 CMTS - BCM3138 Rev A2
C3>
```

show history

Displays a list of recently entered commands.

```
C3>show history
  show memory
  show tech
  show aliases
  show boot
  show calendar
  show class-map
  show clock
  show context
  show exception
  show history
C3#
```

show ip arp

Syntax: **show ip arp [cable 1/0[.s] | fastethernet 0/n[.s] | macaddr | ipaddr]**

Displays the associated MAC and IP addresses for interfaces or addresses, learned through ARP.

Example:

```
C3>show ip arp
Prot Address         Age(min) Hardware Addr  Vlan Type Interface
IP   10.1.176.254    6        00e0.168b.fc89   -  ARPA B#0-FastEthernet 0/0.0
C3>
```

show ip igmp groups

Syntax: **show ip igmp groups**

Shows all IGMP groups held in the C3 IGMP database.

Example:

```
C3> show ip igmp groups
IGMP Connected Group Membership
Group Address     Interface         Uptime      Expires     Last Reporter
239.255.255.254   Ethernet3/1       1w0d        00:02:19    172.21.200.159
224.0.1.40        Ethernet3/1       1w0d        00:02:15    172.21.200.1
224.0.1.40        Ethernet3/3       1w0d        never       171.69.214.251
224.0.1.1         Ethernet3/1       1w0d        00:02:11    172.21.200.11
224.9.9.2         Ethernet3/1       1w0d        00:02:10    172.21.200.155
232.1.1.1         Ethernet3/1       5d21h       stopped     172.21.200.206
C3>
```

show ip igmp interface

Syntax: **show ip igmp interface [cable 1/0[.s] | fastethernet 0/n[.s]]**

Show all IGMP attributes for all IGMP-aware sub-interfaces or for a specific sub-interface.

Example:

```
C3>show ip igmp interface
Cable 1/0.0:
         IGMP is disabled on subinterface
         Current IGMP version is 2
         Interface IGMP joins 0
         Packets dropped:
                Bad checksum or length 0
                IGMP not enabled on subinterface 0
C3>
```

show ip rip

Syntax: **show ip rip [database]**

Displays routing parameters.

See also: “Router Configuration Mode” on page 6-144.

show ip route

Syntax: **show ip route [connected | rip | static | summary]**

Shows IP-related information. The optional parameters are:

(no parameter)
Shows all known routes.

connected
Shows connected networks.

rip
Shows routes learned through RIP.

static
Shows static routes.

summary
Shows a count of all known networks and subnets.

Example:

```
C3>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - ICMP, B - BGP
       E - EGP, G - GGP, O - OSPF, ES - ES-IS, IS - IS-IS

Gateway of last resort is 192.168.253.70 to network 0.0.0.0

     192.168.253.0/24 is subnetted, 1 subnet
C       192.168.253.0/24 is directly connected, FastEthernet 0/0
C3>
```

See also: “ip route” on page 6-87.

show ipc

Displays inter-process communications information. This command is intended only for CMTS debugging use.

show key chain

Displays the configured key chains.

See also: “key chain” on page 6-90.

show memory

Displays current and cumulative memory usage.

```
C3>show memory
 status   bytes    blocks   avg block  max block
 ------ --------- -------- ---------- ----------
current
   free  98231520        5   19646304  98230848
  alloc   2946192     1367       2155        -
cumulative
  alloc   3707728     6254        592        -
C3>
```

show ntp

Displays NTP server details.

Example:

```
C3> show ntp
IP Address    Interval Master   Success /  Attempts Active Offset (s)
63.149.208.50      300 Yes            0 /        35 Yes    Unknown
C3>
```

show snmp

Displays SNMP activity counters.

Example:

```
C3> show snmp

==SNMP information==
        Agent generates Authentication traps: yes
        Silent drops: 0
        Proxy drops: 0
Incoming PDU Counters:
        Total packets: 752
        Bad versions: 0
        Bad community names: 4
        Bad community uses: 1
        ASN parse errors: 0
        Packets too big: 0
        No such names: 0
        Bad values: 0
        Read onlys: 0
        General errors: 0
        Total MIB objects retrieved: 1588
        Total MIB objects modified: 0
        Get requests: 399
        GetNext requests: 348
        Set requests: 1
        Get responses: 0
        Traps: 0
Outgoing PDU Counters:
        Total packets: 802
        Packets too big: 0
        No such names: 6
        Bad values: 0
        General errors: 0
        Get requests: 0
        GetNext requests: 0
        Set requests: 0
        Get responses: 748
        Traps: 54
C3>
```

show terminal

Displays information about the terminal session environment, including the terminal type and command history size.

```
C3>show terminal

Type: ANSI
Length: 54 lines, Width: 80 columns
Status: Ready, Automore on
Capabilities:
Editing is Enabled.
History is Enabled, history size is 10.
```

See also: “terminal” on page 6-14.

show users

Displays active management sessions on the CMTS (serial or telnet).

```
C3>show users
Line    Disconnect Location        User
          Timer
 tty 0  none       serial-port     arris
```

```
*vty 0  0:15:00    192.168.250.80  arris
C3#
```

show version

Displays current software version information (information shown is for illustrative purposes only. Your file names and dates may differ.).

```
C3>show version
ARRIS CLI version .02
Application image: 3.0.1.27, Dec 16, 2003, 18:28:57
BootRom version 2.19
VxWorks5.4.2

System serial number/hostid: 312
WAN/CPU card serial number: 000312
System uptime is 0 weeks, 0 days, 3 hours, 32 minutes

System image file is: Compact Flash - C:\3.0.1.27.bin
2 FastEthernet interface(s)
1 Cable interface(s)
256 MB DDR SDRAM memory

Compact Flash:
        118142976 bytes free,
        9895936 bytes used,
        128038912 bytes total
C3>
```

systat

Identical to the **show users** command.

terminal

Changes the definition of the terminal type, width, or screen length.

```
C3>terminal ?
length             - Set num lines in window
monitor            - Turn on debug output
no                 -
timeout            - Set inactivity timeout period
vt100-colours      - Enable ANSI colours
width              - Set width of window

C3>terminal
```

terminal length

Syntax: **terminal length {n}**

Sets the number of lines that will be displayed before the user is prompted with MORE to continue terminal output. Valid entries of 0 or 2-512 are acceptable

terminal monitor

Syntax: **terminal [no] monitor**

Directs debugging output to the terminal window (the default is to send debug information only to the serial port).

Use the **no** form of this command to stop debugging information from being sent to the current terminal session.

terminal timeout

Syntax: **terminal [no] timeout {n}**

Automatically disconnect terminal sessions if left idle for more than the specified number of seconds (**0** to **65500**). Setting the timeout value to **0**, or using the [**no**] form of this command, disables inactive session disconnection.

terminal vt100-colours

Syntax: **terminal [no] vt100-colours**

Enables or disables ANSI color output.

terminal width

Syntax: **terminal width {n}**

Sets the width of displayed output on the terminal. Valid entries of 1-512 are acceptable.

Privileged Mode Commands

To access commands in privileged mode, use the **enable** command from user mode and enter a valid password.

In privileged mode, the command prompt is the hostname followed by a number sign (e.g., hostname#).

All commands in user mode are valid in privileged mode.

clear ip cache

Syntax: **clear ip cache [ipaddr]**

Clears the route cache for the specified IP address, or the entire cache if no address is specified.

clear ip route

Syntax: **clear ip route [all | rip | static]**

Resets the specified routing table entries.

clear screen

Erases the screen.

configure

Syntax: **configure {terminal | memory | network | overwrite-network}**

Changes the command entry mode to global configuration mode. See “Global Configuration Commands” on page 6-66 for details.

```
C3#configure
Configuring from terminal, memory, or network [terminal]
?t

C3(config)#
```

disable

Exits to user mode.

exit

Close the CMTS connection (same action as **logout**).

help

Displays a brief help listing.

```
C3# help

Press '?' at any time for help on available commands or command
syntax
C3#
```

hostid

Displays the host ID of the C3. Use this to find the proper host ID when ordering feature licenses.

See also: “license” below.

license

Syntax: **license {file name | key n feature ARSVSnnnn | remove n | tftp ipaddr file}**

Enables or removes licensed features on the C3. Contact your ARRIS representative for available features and keys.

Example:

```
C3#license key 0123ABCD456789EF feature ARSVS01163
          RIP            ARSVS01163 enabled
```

See also: “show license” on page 6-60.

logout

Closes the connection to the CMTS regardless of operating mode.

no

Reverses many commands.

show

In privileged mode, displays detailed information about the CMTS configuration. Privileged mode supports the user mode **show** options, and adds the following options.

Type	Name	Page
File System	show c:	6-21
	show file	6-23
	show flash	6-24

Type	Name	Page
Cable Specific	show cable actions	6-HIDDEN
	show cable filter	6-29
	show cable flap-list	6-29
	show cable frequency-band	6-31
	show cable group	6-31
	show cable host	6-31
	show cable modem	6-32
	show cable modulation-profile	6-35
	show cable service-class	6-36
Environment Specific	show access-lists	6-44
	show arp	6-7
	show bridge	6-47
	show bridge-group	6-47
	show cli	6-48
	show configuration	6-49
	show context	6-49
	show controller	6-49
	show debug	6-51
	show environment	6-52
	show interfaces	6-53
	show ip…	6-60
Environment Specific (continued)	show license	6-60
	show logging	6-61
	show mib	6-61
	show processes	6-61
	show reload	6-64
	show running-configuration	6-64
	show snmp-server	6-64
	show startup-configuration	6-64
	show tech-support	6-64

File System Commands

cd

Syntax: **cd {dir}**

Changes the working directory on the Compact Flash disk.

chkdsk

Syntax: **chkdsk {flash: | filesys} [repair]**

Verifies that the file system is correct. The specified *filesys* may be any of the file systems listed by **show file systems**. If the **repair** keyword is specified, the C3 attempts to repair file system errors.

```
C3#chkdsk ?
flash:             - Check flash
<STRING>           - File system

C3#chkdsk flash

Are you sure you want to perform this command?(Y/N)Y
C:/  - disk check in progress ...
C:/  - Volume is OK

          total # of clusters:  62,519
           # of free clusters:  58,117
            # of bad clusters:  0
             total free space:  116,234 Kb
     max contiguous free space:  119,023,616 bytes
                   # of files:  14
                 # of folders:  11
         total bytes in files:  8,758 Ib
             # of lost chains:  0
   total bytes in lost chains:  0
C3#
```

copy

Syntax: **copy {orig} {dest}**

Duplicates the file **orig** and names it **dest**. Specify files by name or use the special qualifiers:

flash
Copy a file on the flash disk to the flash disk or a TFTP server.

running-configuration
Copy the running configuration to a file or the startup configuration.

startup-configuration
Copy the startup configuration to a file or the running configuration.

tftp
Copy a file from the default TFTP server to the flash disk.

tftp://ipaddr/file
Copy a file (or configuration) to or from the TFTP server at the specified address.

If copying to or from the local disk, make sure that the drive letter is in upper case.

Example:

```
C3# copy tftp://10.1.100.1/vxWorks1.st vxWorks1.st

C3#copy C:/test.txt C:/test.old.txt
Copying....!C3#
29886 bytes copied in 0 secs <29886 bytes/sec>
```

delete

Syntax: **delete {filename }**

Removes the specified file from the Compact Flash module.

dir

Syntax: **dir [path]**

Displays a list of all files in the current directory or the specified directory path. Use **show c:** for even more information.

erase

Syntax: **erase {c: | startup-configuration}**

Erases the Flash disk or startup configuration, as specified.

format

Syntax: **format c:**

Completely erases a Compact Flash card and establishes a new file system on it.

mkdir

Syntax: **mkdir {dir}**

Creates a new directory.

more

Syntax: **more {file} [crlf | binary]**

Displays the contents of the specified file, one page at a time. The options are:

no option
displays ignoring missing carriage returns in Unix files

crlf
Properly displays a text file transferred from an MS-DOS or Windows operating system.

binary
Displays a binary file.

Press **c** to display the entire file without pausing, ↵ to view one line at a time, **space** to page down, or **esc** to quit viewing the file.

pwd

Displays the name of the current working directory.

```
C3#pwd
C:/

C3#
```

rename

Syntax: **rename {oldfile} {newfile}**

Changes the name of the file called *oldfile* to *newfile* on the Compact Flash module.

rmdir

Syntax: **rmdir {dir}**

Removes the specified directory. The C3 does not remove an empty directory.

show c:

Syntax: **show c: [all | filesys]**

Displays a complete file listing or optional information about the filesystem on the Compact Flash disk. Use the **filesys** keyword to view the filesystem information; use **all** to display both the file listing and the information (information shown below is for illustrative purposes only. Actual displays will vary).

```
C3#show c:

Listing Directory C::
-rwxrwxrwx  1 0       0            8308 Jul  9 03:01 autopsy.txt
-rwxrwxrwx  1 0       0             996 May 17 00:05 root.der
-rwxrwxrwx  1 0       0           10845 Jul  9 03:01 snmpd.cnf
```

```
-rwxrwxrwx  1 0       0               40 May 17 00:05 tzinfo.txt
-rwxrwxrwx  1 0       0            37623 May 17 00:05 icbImg.txt
-rwxrwxrwx  1 0       0            17177 May 17 00:05 fp_uload.hex
-rwxrwxrwx  1 0       0          2357777 Jul  9 03:00 shutdownDebug.log
-rwxrwxrwx  1 0       0            13023 May 17 00:05 dfu_uload.hex
drwxrwxrwx  1 0       0             2048 May 27 21:33 CONFIG/
drwxrwxrwx  1 0       0             2048 May 17 00:07 SOFTWARE/
-rwxrwxrwx  1 0       0              496 Jun 18 04:49 snmpd.log
-rwxrwxrwx  1 0       0             8112 Jul  9 03:01 snmpd.jnk
-rwxrwxrwx  1 0       0            10845 Jul  9 03:01 snmpd.cnf~
drwxrwxrwx  1 0       0             2048 May 22 09:57 Syslog/
-rwxrwxrwx  1 0       0             8277 Jul  9 03:34 startup-configuration
-rwxrwxrwx  1 0       0             8277 Jul  9 03:34 startup-temp
drwxrwxrwx  1 0       0             2048 May 22 02:34 tftpboot/
-rwxrwxrwx  1 0       0              914 Jun 10 23:10 rootEuro.der
-rwxrwxrwx  1 0       0             1300 Jul  9 03:40 tmp_file-0000

Listing Directory C:/CONFIG:
drwxrwxrwx  1 0       0             2048 May 27 21:33 ./
drwxrwxrwx  1 0       0             2048 Jul  9 03:40 ../
drwxrwxrwx  1 0       0             2048 May 17 00:05 DELETED/
drwxrwxrwx  1 0       0             2048 May 17 00:05 TEMP/
drwxrwxrwx  1 0       0             2048 May 17 00:07 CURRENT/
drwxrwxrwx  1 0       0             2048 May 17 00:07 ALT/

Listing Directory C:/CONFIG/DELETED:
drwxrwxrwx  1 0       0             2048 May 17 00:05 ./
drwxrwxrwx  1 0       0             2048 May 27 21:33 ../

Listing Directory C:/CONFIG/TEMP:
drwxrwxrwx  1 0       0             2048 May 17 00:05 ./
drwxrwxrwx  1 0       0             2048 May 27 21:33 ../

Listing Directory C:/CONFIG/CURRENT:
drwxrwxrwx  1 0       0             2048 May 17 00:07 ./
drwxrwxrwx  1 0       0             2048 May 27 21:33 ../

Listing Directory C:/CONFIG/ALT:
drwxrwxrwx  1 0       0             2048 May 17 00:07 ./
drwxrwxrwx  1 0       0             2048 May 27 21:33 ../

Listing Directory C:/SOFTWARE:
drwxrwxrwx  1 0       0             2048 May 17 00:07 ./
drwxrwxrwx  1 0       0             2048 Jul  9 03:40 ../
drwxrwxrwx  1 0       0             2048 May 17 00:05 DELETED/
drwxrwxrwx  1 0       0             2048 May 17 00:05 TEMP/
drwxrwxrwx  1 0       0             2048 May 17 00:07 CURRENT/
drwxrwxrwx  1 0       0             2048 May 17 00:07 ALT/

Listing Directory C:/SOFTWARE/DELETED:
```

```
drwxrwxrwx  1 0       0             2048 May 17 00:05 ./
drwxrwxrwx  1 0       0             2048 May 17 00:07 ../

Listing Directory C:/SOFTWARE/TEMP:
drwxrwxrwx  1 0       0             2048 May 17 00:05 ./
drwxrwxrwx  1 0       0             2048 May 17 00:07 ../

Listing Directory C:/SOFTWARE/CURRENT:
drwxrwxrwx  1 0       0             2048 May 17 00:07 ./
drwxrwxrwx  1 0       0             2048 May 17 00:07 ../

Listing Directory C:/SOFTWARE/ALT:
drwxrwxrwx  1 0       0             2048 May 17 00:07 ./
drwxrwxrwx  1 0       0             2048 May 17 00:07 ../

Listing Directory C:/Syslog:
drwxrwxrwx  1 0       0             2048 May 22 09:57 ./
drwxrwxrwx  1 0       0             2048 Jul  9 03:40 ../
-rwxrwxrwx  1 0       0            14000 Jun 21 01:59 nvlog.bin

C3#
```

show file

Syntax: **show file {descriptors | systems}**

Lists detailed internal information about file usage, depending on the keyword used. The parameters are:

descriptors
Lists all open file descriptors.

systems
Lists file systems and information about them.

```
C3#show file descriptors
 fd name                   drv
  3 /tyCo/1                  1 in out err
  4 (socket)                 4
  5 (socket)                 4
  6 (socket)                 4
  7 C:/autopsy.txt           3
  8 /snmpd.log               3
  9 (socket)                 4
 10 (socket)                 4
 11 /pty/cli0.M              9
 12 /pty/cli1.M              9
 13 /pty/cli2.M              9
 14 /pty/cli3.M              9
 15 /pty/cli4.M              9
 16 /pty/cli0.S              8
 17 /pty/cli1.S           8
 18 /pty/cli2.S           8
 19 /pty/cli3.S           8
 20 /pty/cli4.S           8
 21 (socket)              4
 22 (socket)              4
C3#

C3#show file systems
drv name
  0 /null
  1 /tyCo/1
  3 C:
  5 Phoenix1:
  7 /vio
  8 /pty/cli0.S
  9 /pty/cli0.M
  8 /pty/cli1.S
  9 /pty/cli1.M
  8 /pty/cli2.S
  9 /pty/cli2.M
  8 /pty/cli3.S
  9 /pty/cli3.M
  8 /pty/cli4.S
  9 /pty/cli4.M
C3#
```

show flash

Syntax: **show flash [all | filesys]**

Displays detailed information about the Compact Flash disk, depending on the option used. The options are:

(no option)

Display Files and directories only (identical to the **show c:** command).

all

Display all files, directories and filesystem detail.

filesys

Display only filesystem detail.

Example:

```
C3#show flash filesys

==== File system information ====

volume descriptor ptr (pVolDesc):       0x89ecf4f0
cache block I/O descriptor ptr (pCbio): 0x89ecf7dc
```

```
auto disk check on mount:               DOS_CHK_REPAIR | DOS_CHK_VERB_SILENT
max # of simultaneously open files:     22
file descriptors in use:                2
# of different files in use:            2
# of descriptors for deleted files:     0
# of  obsolete descriptors:             0

current volume configuration:
 - volume label:        NO NAME ; (in boot sector:      NO NAME    )
 - volume Id:           0x163317f2
 - total number of sectors:     250,592
 - bytes per sector:            512
 - # of sectors per cluster:    4
 - # of reserved sectors:       1
 - FAT entry size:              FAT16
 - # of sectors per FAT copy:   245
 - # of FAT table copies:       2
 - # of hidden sectors:         32
 - first cluster is in sector # 523
 - directory structure:         VFAT
 - root dir start sector:               491
 - # of sectors per root:               32
 - max # of entries in root:            512

FAT handler information:
------------------------
 - allocation group size:       7 clusters
 - free space on volume:        127,891,456 bytes
C3#
```

write

Syntax: **write [memory | terminal | network file | erase]**

Writes the running configuration, or erases the startup configuration, based on the argument. The options are:

(no option)
Saves the running configuration to the startup configuration (to disk).

memory
Saves the running configuration to the startup configuration (to disk).

terminal
Displays the running configuration on the terminal.

network
Saves the running configuration to the specified file. The file may be a path on the Compact Flash disk, or you can specify

tftp://n.n.n.n/filename to copy the configuration to a TFTP server.

erase

Erases the startup configuration on the Compact Flash disk. If you do no create a new startup configuration, the CMTS uses the factory default configuration at the next reload. See also “Bridge Groups” on page 3-4.

Cable Specific Commands

The following commands affect or display the status of attached cable modems. These commands are available only in privileged mode.

cable modem

Syntax: **[no] cable modem {address} {max-hosts n | subscriber {auto}}**

Sets user and QoS parameters. The parameters are:

address

Specify a cable modem by IP address, MAC address, or **all** to specify all cable modems on the CMTS.

max-hosts

Sets the maximum number of CPE devices allowed to communicate through the cable modem. Use the keyword **default** to specify the default number of devices.

subscriber

Adds the specified static IP address to the list of valid subscribers.

auto

Automatically learn the subscriber's IP address.

clear cable flap-list

Syntax: **clear cable flap-list {all | macaddr}**

Clear the flap list for all modems or for the modems with the specified MAC address.

Example:

```
C3#scm

I/F      Prim Online     Timing Rec    CPE   IP Address      MAC Address    DOC
         SID  State      Offset Power                                       Mode
C1/0/U1  1    online     3167   -4.7   0/1   10.99.88.100    00a0.731e.3f84 D1.0


C3#clear cable flap-list 00a0.731e.3f84
C3#
```

clear cable modem

Syntax: **clear cable modem {all | ipaddr | macaddr | offline} {reset | counters | delete}**

Resets, removes, or deletes the specified cable modems. The parameters are:

all

Specify all cable modems.

ipaddr

Specify the modem by IP address.

macaddr

Specify the modem by MAC address.

offline

Specify offline modems. Valid only when used with the **delete** subcommand.

reset

Reboots the specified modems. This is accomplished by sending the modem a ranging message with the “Abort” flag set. In addition, the C3 removes the modem from the ranging list, which should result in the modem rebooting within 30 seconds per the DOCSIS specification. when a modem is reset, the upstream channel associated with that modem is still known and is displayed.

counters

Clears all counters associated with the specified modems.

delete

Resets the specified modems and removes them from the CMTS database.

Example (showing cable modem cleared from ranging list):

```
C3#show cable modem

I/F     Prim Online     Timing Rec    CPE IP Address      MAC address    DOC
        SID  State      Offset Power                                     Mode
C1/0/U0 1    online     3165   -3.0   -   192.168.253.67  00a0.731e.3f84 D1.0

C3#clear cable modem 192.168.253.67 reset
Cable modem 192.168.253.67 has been reset

C3#show cable modem

I/F     Prim Online     Timing Rec    CPE IP Address      MAC address    DOC
        SID  State      Offset Power                                     Mode
```

```
C1/0/U0 0    offline    0      0.0    -   0.0.0.0         00a0.731e.3f84 D1.0

C3#

or

C3#scm

I/F     Prim Online     Timing Rec    CPE IP Address      MAC address    DOC
        SID  State      Offset Power                                     Mode
C1/0/U0 1    online     3160   -3.0   -   192.168.253.67  00a0.731e.3f84 D1.0

C3#clear cable modem 00a0.731e.3f84 reset
Cable modem 00a0.731e.3f84 has been reset
C3#scm
I/F     Prim Online     Timing Rec    CPE IP Address      MAC address    DOC
        SID  State      Offset Power                                     Mode
C1/0/U0 0    offline    0      0.0    -   0.0.0.0         00a0.731e.3f84 D1.0

C3#
C3#clear cable modem all reset
Total modems = 9,       Online= 8, offline = 1
Total reset = 8
C3#
```

See also: “cable modem offline aging-time” on page 6-75.

clear logging

Clears the local event log.

show cable filter

Syntax: **show cable filter [group gid] [verbose]**

Lists filters configured on the selected cable modems.

group

Specifies the group ID. Valid range: **1** to **30**. If you do not specify a group, the C3 shows all configured groups.

verbose

Prints a more detailed listing.

See also: “cable filter group” on page 6-69, “cable filter” on page 6-69, “cable submgmt default filter-group” on page 6-82.

show cable flap-list

Syntax: **show cable flap-list [cable x/y | settings | sort-flap | sort-interface | sort-mac | sort-time | summary]**

Displays the current contents of the flap list. The following options restrict or sort output:

(no option)
sort-flap

Sort by flap count (default).

settings

Lists the current flap list data accumulation settings. The columns in the report are:

Column	Description
Flap aging time	Aging time in days of cable modem flap events.
Flap insertion Time:	If a modem is online less than this time (seconds), the CMTS records the modem in the flap list.
Flap Miss Threshold	The number of times a modem can miss the background keep alive polling before being listed as a flap event.
Power adjustment threshold	The power level change that triggers a flap event for a modem.
Flap list size	Number of entries recorded in the flap list.

sort-interface

Sort by interface.

sort-mac

Sort by MAC address.

sort-time

Sort by time.

cable x/y

Show the flap list for a specific cable interface.

Example:

```
Mac Addr       CableIF Ins   Hit   Miss  CRC   Flap  Time
0090.836b.452d C1/0/U0 1384  7     0     12    1385  NOV 25 18:26:29
00a0.7300.0012 C1/0/U4 711   5     0     0     711   NOV 25 22:08:56
00a0.7312.4bd8 C1/0/U4 449   100   23    0     621   NOV 25 22:19:01
00a0.7312.4be9 C1/0/U4 361   70    4     0     549   NOV 25 22:02:33
00a0.7312.4c7b C1/0/U4 307   91    0     0     522   NOV 24 06:14:14
00a0.7312.4c1f C1/0/U5 145   21    23    0     509   NOV 24 06:10:44
00a0.7388.9167 C1/0/U4 5     2284  1525  179   288   NOV 25 22:20:22
00a0.7316.6a2e C1/0/U5 180   0     0     0     180   NOV 23 01:56:34
```

```
00a0.7311.43fe C1/0/U4 124   48    0    0    124   NOV 23 01:44:11
00a0.73ad.3827 C1/0/U2 5     21179 1354 0    43    NOV 23 15:25:35
00a0.7314.2ecc C1/0/U4 0     26546 27   0    29    NOV 25 18:48:12

C3#show cable flap-list summary
show cable flap-list: print per/upstream summary

CableIF Ins       Hit       Miss      CRC       Flap
C1/0/U0 597       22605     3320      16        1029
C1/0/U2 5         111       87        3         13
C1/0/U3 46        77        160       0         56
C1/0/U4 16        0         0         0         16
C1/0/U5 94        86        238       14        130

C3#show cable flap-settings

Flap      Flap      Range     Power     Flap
Aging     Insertion Miss      Adjust    List
Time      Time      Threshold Threshold Size
10        180       6         3         500
```

show cable frequency-band

Syntax: **show cable frequency-band [index]**

Displays the specified frequency group, or all frequency groups if no frequency group is specified.

See also: “cable frequency-band” on page 6-73.

show cable group

Syntax: **show cable group [n]**

Displays the selected cable group and its load balancing configuration. Specify no option to display all configured cable groups.

show cable host

Syntax: **show cable host {ipaddr | macaddr}**

Displays all CPE devices connected to the cable modem, specified by IP address or MAC address. Host IP address only returned if subscriber management is turned on. The information is returned using the C3 knowledge of active CPE behind the specified modem and not by using an SNMP query on the modem. The parameters are:

ipaddr
IP address of modem to view.

macaddr
MAC address of modem to view.

See also: “show interfaces cable 1/0 modem” on page 6-56, “cable sub-mgmt…” on page 6-80.

show cable modem

Syntax: **show cable modem [ipaddr | macaddr | cable 1/0 [upstream n]] [detail | offenders | registered | summary | unregistered | columns cols|snr] [count] [verbose]**

Displays information about the specified cable modem, or all registered cable modems if no modem is specified. The options are:

cable 1/0
View all modems on the cable interface (options limited to **registered** and **unregistered**).

cable 1/0 upstream [n]
View all modems on the specified upstream (options limited to **registered** and **unregistered**). Valid range: **0** to **5**.

detail
Displays information including the interface that the modem is acquired to, the SID, MAC, concatenation status, and the received signal-to-noise ratio.

ipaddr
Optional IP address of modem to view.

macaddr
Optional MAC address of modem to view.

offenders
Show top cable modems for packets throttled or spoofing.

registered
Displays registered modems (**online** or **online(pt)**) and does not display the earlier states. All states are displayed by **show cable modem** without any modifiers.

summary
Displays the total number of modems, the number of active modems, and the number of modems that have completed registration.

unregistered
Displays modems which have ranged but not yet registered (including offline modems).

count
Specify a maximum number of cable modems to display.

verbose

Provide additional information.

columns

Show selected columns (one or more, separated by spaces) from the following list. Allows customization of output.

Column Name	Description
CORRECTED-FEC	Corrected FEC Codewords
CPE	CPE information
GOOD-FEC	Good FEC Codewords
INTERFACE	Interface
IP	IP address
MAC	MAC address
PROV-MODE	Provisioned mode
REC-PWR	Receive Power
REG-TYPE	Registration Type
SID	Prim
SNR	Signal to Noise Ratio
STATUS	Status
TIMING	Timing offset
UNCORRECTED-FEC	Uncorrected FEC Codewords
UP-MOD	Upstream Modulation
VLAN-BGROUP	VLAN ID

See also: “show interfaces cable 1/0 modem” on page 6-56.

Example (**detail**):

```
C3#show cable modem detail

MAC Address                     : 00a0.731e.3f84
IP Address                      : 10.99.88.100
Primary SID                     : 1
Interface                       : C1/0/U1
Timing Offset                   : 3167
Received Power                  : -4.7 dBmV (SNR = 66.3 dBmV)
Provisioned Mode                : D1.0
Registration Type               : D1.0
Upstream Modulation             : TDMA
Ranging/Registration            : online - BPI not enabled
Total good FEC CW               : 377
Total corrected FEC             : 0
Total uncorrectable FEC       : 0

C3#
```

Example (**registered**):

```
C3#show cable modem registered

I/F      Prim Online     Timing Rec    CPE   IP Address      MAC Address    DOC
         SID  State      Offset Power                                       Mode
C1/0/U1  1    online     3167   -4.7   0/1   10.99.88.100    00a0.731e.3f84 D1.0
C3#
```

The **show cable modem registered** command reports one of the following states for each modem:

State	Meaning
Offline	The cable modem is inactive.
init(r1)	The C3 has successfully received a ranging request from the modem in a contention interval (i.e., initial ranging)
init(r2)	The CMTS has responded to an initial ranging request from the modem, but has not yet completed ranging (i.e., the modem's transmit parameters are still outside of the acceptable range as defined by the CMTS).
init(rc)	The cable modem has successfully adjusted its transmit power and timing so that initial ranging has completed successfully.
init(d)	The cable modem has sent a DHCP request.
init(o)	The modem is ready to or is currently TFTP'ing the configuration file.
init(t)	modem ready for ToD
Online	The modem has successfully completed registration.
Online(d)	online, network access disabled
Online(pt)	The modem is online and BPI is enabled. The modem has a valid traffic encryption key (TEK).
Online(pk)	The modem is online, BPI is enabled, and a key encryption key (KEK) is assigned.
reject(m)	The CMTS rejected the registration request from the modem because the shared secret from the modem does not match the CMTS shared secret.
reject(c)	The class of service offered by the modem as part of the registration request was not valid.
reject(pk)	The Key Encryption Key (KEK) offered by the modem was invalid.

State	Meaning
reject(pt)	The Traffic Encryption Key (TEK) offered by the modem was invalid.

Example (**summary**):

```
C3#show cable modem sum

Interface  Total Offline Unregistered Rejected Registered

Cable1/0/U0 1    0       0            0        1
Cable1/0/U1 0    0       0            0        0
Cable1/0    1    0       0            0        1
```

Example (**summary verbose**):

```
C3#show cable modem sum verbose

Interface  Total Offline Ranging    Ranging      IP        Rejected Registered
                                 Aborted|Completed Completed
Cable1/0/U0 1    0       0       0       0         0         0        1
Cable1/0/U1 0    0       0       0       0         0         0        0
Cable1/0    1    0       0       0       0         0         0        1

C3#
```

Example (**columns**):

```
C3#show cable modem columns IP MAC VLAN
IP address      MAC address    Vlan
                               ID
0.0.0.0         00a0.73ae.ec13 3
0.0.0.0         00a0.7374.b99e 4
C3#
```

show cable modulation-profile

Syntax: **show cable modulation-profile [advphy | n [type] [verbose]]**

Displays information about the specified modulation profile, or all profiles if none is specified. The parameters are:

advphy

Shows TDMA and SCDMA parameters for each modulation profile and IUC type.

n

The modulation profile to display. Valid range: **1** to **10**.

type

The IUC type; one of: **advphy**, **advphyl**, **advphys**, **advphyu**, **initial**, **long**, **reqdata**, **request**, **short**, **station**.

verbose

Show profile parameters in a list format. The default is to show parameters in a table format with abbreviated parameter names.

Example (showing the factory default profile):

```
C3#show cable modulation-profile 1

Mod IUC     Type Preamb Diff FEC    FEC  Scrambl Max  Guard Last Scrambl
                 length enco T      CW   Seed    B    time  CW
                             BYTES  SIZE         size size  short
1   request qpsk  64    no   0x0    0x10  0x152  0    8     no   yes
1   initial qpsk  640   no   0x5    0x22  0x152  0    48    no   yes
1   station qpsk  384   no   0x5    0x22  0x152  0    48    no   yes
1   short   qpsk  84    no   0x6    0x4e  0x152  13   8     no   yes
1   long    qpsk  96    no   0x8    0xdc  0x152  0    8     no   yes
1   advPhyS 64qam 104   no   0xc    0x4b  0x152  6    8     no   yes
1   advPhyL 64qam 104   no   0x10   0xdc  0x152  0    8     no   yes
C3#
```

show cable service-class

Syntax: **show cable service-class [verbose]**

Displays defined service classes. Use the **verbose** keyword to see a more detailed listing.

Example:

```
C3#show cable service-class
Name            State Dir Sched Prio MaxSusRate MaxBurst   MinRsvRate
test             Act   US  BE     0    200000     3044       0
Multicast        Inact DS  BE     0    0          0          0
basic_upstream   Act   US  BE     0    0          3044       0
```

Environment Specific Commands

calendar set

Syntax: **calendar set {hh:mm:ss} [dd mmm yyyy]**

Sets the internal CMTS real time clock to the specified time. The calendar keeps time even if the CMTS is powered off.

Example:

```
C3#calendar set 13:59:11 02 sep 2003
```

clear access-list

Syntax: **clear access-list counters [n]**

Clears the counters on the specified access list, or all access lists if no list is specified.

See also: “show access-lists” on page 6-44, “access-list” on page 6-66.

clear arp-cache

Clears the ARP cache.

See also: “show ip arp” on page 6-10, “show arp” on page 6-7.

clear ip igmp group

Syntax: **clear ip igmp group [ipaddr]**

Deletes the specified IGMP group from the multicast cache, or all IGMP groups if none is specified. The IP address range is **224.0.0.0** to **239.255.255.255**.

See also: “show ip igmp groups” on page 6-10.

clear mac-address

Syntax: **clear mac-address {macaddr}**

Deletes the learned MAC address entry from the table.

clear mac-address-table

Deletes all learned entries from the MAC address table.

clock set

Syntax: **clock set {hh:mm:ss} [dd MMM yyyy]**

Sets the CMTS clock to the specified time (and optionally, date). The CMTS synchronizes the clock to the CMTS calendar when powered on or rebooted.

```
C3# clock set 13:59:11 05 feb 2004
```

debug

Syntax: **[no] debug**

Enables debugging output to the serial console (or telnet sessions if the **term monitor** command is used in a telnet session).

Debug commands are global across terminal and telnet sessions. Use the **terminal monitor** command to send debug output to a telnet session. Debug may be enabled in one telnet session and disabled in another telnet session. Use **show debug** to show the state of debugging across all sessions.

CAUTION
Reduced system performance
Producing debugging information can consume extensive CMTS resources, which may result in reduced system performance. For best results, only enable debugging when necessary and disable it as soon as it is no longer needed.

To turn off debugging, give the command **no debug** or **undebug**.

Debugging can be turned on and off (the **no** form of the command) for one or many modems based on MAC address or primary SID. Modems are added to the debug list when specified and removed with the **no** command variant.

Commands that add/remove modems from the debug list are:

```
[no] debug cable interface <type x/y> [
     [mac-address <M.M.M> [m.m.m] ] | sid <nnnn> ] [verbose]
[no] debug cable mac-address <M.M.M> [m.m.m] [verbose]
[no] debug cable sid <NNNN> [verbose]
```

Use the **show debug** command to see what modems are in the debug list:

```
C3#show debug

Mac Addresses enabled for Debug:



Primary Sids enabled for Debug:



Debugging events/message types which are enabled:

Contents of Cable Modem Database debuglevel:
I/F     PrimSid   MAC address     Debug

C3#
```

debug all

Syntax: **[no] debug all**

Provides all debugging information.

Use **no debug all** to turn off debug for all cable modems for all events.

Use **debug all** to turn on debug in terse mode for all cable modems previously being debugged.

debug cable dhcp-relay

Syntax: **[no] debug cable dhcp-relay**

Enables or disables DHCP relay debugging.

debug cable interface

Syntax: **[no] debug cable interface cable 1/0 {mac-address macaddr [macmask] | sid n} [verbose]**

Enable or disable debugging on the selected cable modem or interface. The options are:

mac-address

Enables debugging on the cable modem with the specified MAC address. If the optional mask is included, the CMTS enables debugging on all cable modems whose MAC address, AND'ed with the mask, matches the specified MAC address.

sid

Enables debugging on the cable modem with the specified Service ID (SID).

verbose

Enables verbose debugging. The CMTS defaults to terse mode.

debug cable mac-address

Syntax: **[no] debug cable mac-address {macaddr} [mask] [verbose]**

Enables or disables debugging on the cable modems matching the specified MAC address. The options are:

macaddr

Enables debugging on the cable modem with the specified MAC address.

mask

Enables debugging on all cable modems whose MAC address, AND’ed with the mask, matches the specified MAC address.

verbose

Enables verbose debugging. The CMTS defaults to terse mode.

debug cable privacy

Syntax: **[no] debug cable privacy [mac-address macaddr] [level n]**

Enables Baseline Privacy (BPI) debugging on the specified cable modem. The options are:

macaddr

The MAC address of the cable modem.

level

The BPI debug level:

0—no output

1—trace incoming/outgoing messages

2—same as level 1 and display information of incoming message

3—same as level 2 and display outgoing message data

debug cable range

Syntax: **[no] debug cable range**

Enables ranging debug messages for all cable modems.

debug cable registration

Syntax: **[no] debug cable registration**

Enables modem registration request debug messages.

debug cable sid

Syntax: **[no] debug cable sid {NNN} [verbose]**

Enables debugging on the cable modem with the specified primary SID.

debug cable tlvs

Syntax: **[no] debug cable tlvs**

Enables Type-Length Value (TLV) debugging messages.

debug envm

Syntax: **[no] debug envm**

Enables environment debugging messages.

debug ip

Syntax: **[no] debug ip [rip]**

Enables debuggin messages. The options are:

rip

Enables RIP debugging messages.

```
C3#debug ip
RIP protocol debugging is on
!Note: this debug message typde is non-blocking and some     messages
may be lost if the system is busy
!Note: debug messages of this type can only be displayed on teh
    console, not on telnet sessions

C3#debug ip rip
RIP protocol debugging is on
!Note”: this debug message ytpe is non-blocking and some     messages
may be lost if the system is busy
```

debug snmp

Syntax: **[no] debug snmp**

Enables debug messages for SNMP.

debug syslog

Syntax: **[no] debug syslog**

Enables debug messages for Syslog traffic.

debug telnet

Syntax: **[no] debug telnet**

Enables debug messages for incoming telnet sessions.

disable

Exits privileged mode, returning the session to user mode.

```
C3#disable
C3>
```

disconnect

Syntax: **disconnect vty {id}**

Disconnects telnet sessions even if not fully logged in yet. Valid range: **0** to **3**.

Example:

```
C3#show user
Line    Disconnect Location        User
          Timer
*tty 0  0:14:57    serial-port     arris
 vty 0  0:15:00    192.168.250.80  arris
 vty 1  0:15:00    192.168.250.80  arris
 vty 2  0:15:00    192.168.250.80  arris
 vty 3  0:15:00    192.168.250.80  arris
C3#disconnect vty 2
```

login

Syntax: **login user {name str | password str}**

Changes the user level login name and password for telnet sessions.

Example:

```
C3#login user name arris
C3#login user password arris
C3#
```

See also: “Initial Configuration” on page 2-12 to set the password for privilege access level.

ping

Syntax: **ping {ipaddr}**

Pings the specified IP address.

Example:

```
C3#ping 192.168.253.66
PING 192.168.253.66: 56 data bytes
64 bytes from Phoenix1 (192.168.253.66): icmp_seq=0. time=0. ms
64 bytes from Phoenix1 (192.168.253.66): icmp_seq=1. time=0. ms
64 bytes from Phoenix1 (192.168.253.66): icmp_seq=2. time=0. ms
64 bytes from Phoenix1 (192.168.253.66): icmp_seq=3. time=0. ms
64 bytes from Phoenix1 (192.168.253.66): icmp_seq=4. time=0. ms
----192.168.253.66 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
C3#
```

reload

Syntax: **reload [at time [reason] | cancel | in time [reason]]**

Restarts the CMTS (same behavior as setting **docsDevResetNow** to **true**). The parameters are:

at

Specifies the clock time, in **hh:mm** notation, to reboot the C3. You can add an optional reason string, describing why the reboot was necessary.

in

Specifies the amount of time, in **hh:mm** notation, to wait before rebooting the C3. You can add an optional reason string, describing why the reboot was necessary.

cancel

Cancels a scheduled reboot.

The CMTS prompts you to save the running configuration to the startup configuration if changes to the configuration have been made. If you choose not to save the running configuration to the startup configuration, the CMTS appends a copy of the running configuration to the shutdowndebug.log file on the Compact Flash disk.

Example (entering **N** for the confirmation):

```
C3#reload
Proceed with reload? (Y/N)

Operation Cancelled!
C3#
```

script start

Syntax: **script start {*file*}**

Starts recording a command script to the specified file.

script execute

Syntax: **script execute {*file*}**

Executes a recorded script in the specified file.

script stop

Finishes recording a command script.

send

Syntax: **send {all | console | vty0 | vty1 | vty2 | vty3} {message}**

Sends a text message to the specified CLI users.

```
C3#send all "testing"
```

***

```
***
*** Message from  vty0 to all terminals:
***
testing




C3#
```

show access-lists

Syntax: **show access-lists [acl | interface matches | cable X/Y.Z matches| fastethernet X/Y.Z matches]**

Displays access-list information. It can be supplied with an access-list-number. Implicit ACE, ACE index and ACL type (extended/standard) is shown in output. The options are:

(no option)

Displays the full list of configured ACLs.

acl

Displays the specified ACL configuration.

interface matches|cable matches|fastethernet matches

Displays statistics of matches against each interface in each direction. “Interface cable X/Y.Z matches” or “interface fasthernet X/Y.Z” shows ACL's for the selected sub-interface.

Example (single ACL):

```
C3>show access-lists 1

access-list 1 permit 192.5.34.0  0.0.0.255
access-list 1 permit 128.88.0.0  0.0.255.255
access-list 1 permit 36.0.0.0  0.255.255.255
! (Note: all other access implicitly denied
>

C3>show access-lists

Extended IP access list 100
        [01] permit ip any any <matches 00>
        DEFAULT deny ip any any <matches 00>
>
```

Example (no option, display the full list):

```
C3#show access-lists
Extended IP access list 2699
        [01] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
priority (matches 0)
```

```
        [02] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
immediate (matches 0)
        [03] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
flash (matches 0)
        [04] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
flash-override (matches 0)
        [05] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
critical (matches 25)
        [06] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
internet (matches 547)
        [07] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos 5 precedence
network (matches 0)
        [08] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence network (matches 0)
        [09] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence priority (matches 0)
        [10] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence immediate (matches 0)
        [11] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence flash (matches 0)
        [12] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence flash-override (matches 0)
        [13] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence critical (matches 0)
        [14] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
min-monetary-cost precedence internet (matches 765)
        [15] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
max-reliability precedence network (matches 0)
        [16] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
max-reliability precedence priority (matches 0)
        [17] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
max-reliability precedence immediate (matches 0)
        [18] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
max-reliability precedence flash (matches 125)
        [19] permit tcp host 1.1.1.2 eq 1 host 4.4.4.4 eq 5 tos
max-reliability precedence flash-override (matches 0)
        [20] deny ip any any (matches 43584779)
```

Example (**interface matches**):

```
C3#show access-lists interface matches
Interface               Direction         Acl ID        Entry No.Matches
FastEthernet 0/0.0      Outgoing              78        None Set     N/A
FastEthernet 0/0.0      Inbound             2699               1     0
FastEthernet 0/0.0      Inbound             2699               2     0
FastEthernet 0/0.0      Inbound             2699               3     0
FastEthernet 0/0.0      Inbound             2699               4     0
FastEthernet 0/0.0      Inbound             2699               5     0
FastEthernet 0/0.0      Inbound             2699               6     0
FastEthernet 0/0.0      Inbound             2699               7     0
FastEthernet 0/0.0      Inbound             2699               8     0
FastEthernet 0/0.0      Inbound             2699               9     0
FastEthernet 0/0.0      Inbound             2699              10     0
FastEthernet 0/0.0      Inbound             2699              11     0
FastEthernet 0/0.0      Inbound             2699              12     0
FastEthernet 0/0.0      Inbound             2699              13     0
FastEthernet 0/0.0      Inbound             2699              14     0
FastEthernet 0/0.0      Inbound             2699              15     0
FastEthernet 0/0.0      Inbound             2699              16     0
FastEthernet 0/0.0      Inbound             2699              17     0
FastEthernet 0/0.0      Inbound             2699              18     0
FastEthernet 0/0.0      Inbound             2699              19     0
FastEthernet 0/0.0      Inbound             2699              20     45057477
FastEthernet 0/1.0      Outgoing         Not Set        None Set     N/A
FastEthernet 0/1.0      Inbound             2698               1     0
FastEthernet 0/1.0      Inbound             2698               2     0
FastEthernet 0/1.0      Inbound             2698               3     0
FastEthernet 0/1.0      Inbound             2698               4     0
FastEthernet 0/1.0      Inbound             2698               5     0
FastEthernet 0/1.0      Inbound             2698               6     0
FastEthernet 0/1.0      Inbound             2698               7     0
FastEthernet 0/1.0      Inbound             2698               8     0
FastEthernet 0/1.0      Inbound             2698               9     0
FastEthernet 0/1.0      Inbound             2698              10     0
FastEthernet 0/1.0      Inbound             2698              11     0
FastEthernet 0/1.0      Inbound             2698              12     0
FastEthernet 0/1.0      Inbound             2698              13     0
FastEthernet 0/1.0      Inbound             2698              14     0
FastEthernet 0/1.0      Inbound             2698              15     0
FastEthernet 0/1.0      Inbound             2698              16     0
FastEthernet 0/1.0      Inbound             2698              17     0
FastEthernet 0/1.0      Inbound             2698              18     38772
FastEthernet 0/1.0      Inbound             2698              19     0
FastEthernet 0/1.0      Inbound             2698              20     304
Cable 1/0.0             Outgoing             171               1     0
Cable 1/0.0             Outgoing             171               2     0
Cable 1/0.0             Outgoing             171               3     0
Cable 1/0.0             Outgoing             171               4     0
Cable 1/0.0             Outgoing             171               5     0
Cable 1/0.0             Outgoing             171               6     1529
Cable 1/0.0             Outgoing             171               7     1482
Cable 1/0.0             Outgoing             171               8     186184
Cable 1/0.0            Inbound             2601        None Set     N/A
```

Example (**interface cable 1/0.0 matches)**

```
C3<config>#show access-lists interface cable 1/0.0 matches
Interface               Direction         Acl ID        Entry No.
    Matches
```

```
Cable 1/0.0            Outgoing          Not Set      None Set
     N/A
Cable 1/0.0            Inbound           Not Set      None Set
     N/A
C3<config>#
```

Example (**interface fastethernet 0/0.0 matches)**

```
C3<config>#show access-lists interface cable 1/0.0 matches
Interface              Direction         Acl ID        Entry No.
    Matches
Fastethernet 0/0.0     Outgoing          Not Set      None Set
     N/A
Fastethernet 0/0.0     Inbound           Not Set      None Set
     N/A
C3<config>#
```

show bridge

Displays information from the bridge MIB.

Example:

```
C3#show bridge

        Bridge Address = 0000.ca3f.63ca
        Number of Ports = 3
        Bridge Type = transparent-only
        Learning Discards = 0
        Aging Time(seconds) = 15000

 = Bridge forwarding table =
-MAC Address-    -CMTS Port-         -Status- -Bridge Grp-  -VLAN Tags-
0000.92a7.adcc   FastEthernet 0/0.0   Learned   0            Untagged
0000.ca31.67d3   Cable 1/0.0          Learned   0            Untagged
0000.ca31.6bf9   Cable 1/0.0          Learned   0            Untagged
0000.ca3f.63ca   FastEthernet 0/0     Self      N/A          N/A
0000.ca3f.63cb  *FastEthernet 0/1     Self      N/A          N/A  *NON-OPER
0000.ca3f.63cc   Cable 1/0            Self      N/A          N/A
0001.5c20.4328   FastEthernet 0/0.0   Learned   0            Untagged
C3#
```

show bridge-group

Syntax: **show bridge-group [n]**

Shows details of the specified bridge group, or all bridge groups if you specify no bridge group.

Example:

```
C3(config)#sh bridge-g 1

bridge-group #1: ATTACHED
        Cable 1/0.1
                VLAN-tag #42 (native)
        FastEthernet 0/1.1 - not bridging (no VLAN-tag configured)
        FastEthernet 0/0.1
                VLAN-tag #42

C3(config)#
C3(config)# bridge 1 bind cable 1/0.1 28 fastethernet 0/0.1 44
C3(config)# bridge 1 bind cable 1/0.1 19 fastethernet 0/0.1 83
C3(config)# bridge 1 bind cable 1/0.1 73 fastethernet 0/1.1 53
C3(config)#sh bridge-gr 1

bridge-group #1: ATTACHED
        Cable 1/0.1
                VLAN-tag #42 (native)
               VLAN-tag #19 bound to FastEthernet 0/0.1 VLAN-tag #83
               VLAN-tag #28 bound to FastEthernet 0/0.1 VLAN-tag #44
               VLAN-tag #73 bound to FastEthernet 0/1.1 VLAN-tag #53
        FastEthernet 0/1.1
                VLAN-tag #53 bound to Cable 1/0.1 VLAN-tag #73
        FastEthernet 0/0.1
                VLAN-tag #42
                VLAN-tag #44 bound to Cable 1/0.1 VLAN-tag #28
                VLAN-tag #83 bound to Cable 1/0.1 VLAN-tag #19
```

The following example shows a cable sub-interface with an IP address but as this sub-interface has no encapsulation, specification is “not attached:.

```
C3(config)#ip routing
C3(config)#int cable 1/0.4
!NOTE: sub-interface config will not be applied
!  (and will not be displayed by the “show” commands)
!  until after interface-configuration mode has been exited

C3(config-subif)# ip address 10.99.87.1 255.255.255.0
C3(config-subif)# exit

C3(config)# show bridge-group

bridge-group #4: NOT ATTACHED
        Cable 1/0.4
                         10.99.87.1/24
C3(config)#
```

See also: “bridge” on page 6-67, “bridge-group” on page 6-111, “bridge <n> bind” on page 6-68, “encapsulation dot1q” on page 6-111.

show cli

Displays CLI information.

show cli accounts

Shows login and password strings.

Example:

```
C3#show cli accounts
Login name          : arris
Login password      : arris
Enable password     : arris
Enable secret       :
---------------------
C3#
```

show cli logging

Syntax: **show cli logging [session n]**

Shows global logging information. Specify a user session (**0** to **4**) to display logging information for only one session; no specification displays the global logging parameters.

Example:

```
C3#show cli logging
CLI command logging is: disabled
        logging of passwords is: disabled
        File path for password logging: /

        Max file size: 1024 Kilobytes

C3#
```

show configuration

See “show running-configuration” on page 6-64.

show context

Displays context info about recent crashes.

show controller

Syntax: one of:
show controller cable [x/y]
show controller fastethernet [x/y]
show controller loopback [interface number]

Displays information about the specified interface (or all interfaces if none are specified).

Examples:

```
C3#show controller cable 1/0
```

```
Cable1/0 downstream
        Frequency 681.0 MHz,Channel-Width 6.0 MHz,Modulation 64-QAM
        Power 45.0 dBmV, R/S Interleave I=32, J=4
        Downstream channel ID: 1
        Dynamic Services Stats:
                DSA: 0  REQs  0 RSPs  0 ACKs
                0 Successful DSAs  0 DSA Failures
                DSC: 0  REQs  0 RSPs  0 ACKs
                0 Successful DSCs  0 DSC Failures
                DSD: 0  REQs  0 RSPs
                0 Successful DSDs  0 DSD Failures

                DCC: 0  REQs  0 RSPs  0 ACKs
                0 Successful DCCs  0 DCC Failures

Cable1/0 Upstream 0
        Frequency 10.0 MHz,Channel-Width 3.200000 MHz
        Channel-type: TDMA
        SNR 37.9 dB
        Nominal input power-level -4.0 dBmV(fixed), Tx Timing offset 1964
        Ranging backoff (Configured- Start 16, End 16)(Actual- Start 0, End 2)
        Ranging Insertion Interval (Configured 0 ms) (Actual 1280 ms)
        Tx backoff (Start 0, End 5)
        Modulation Profile Group 1
        Ingress-cancellation is disabled
        Minislot Size in number of Timebase Ticks is = 4
        Upstream channel ID: 1

Cable1/0 Upstream 1
        Frequency 15.0 MHz,Channel-Width 3.200000 MHz
        Channel-type: TDMA
        SNR 0.0 dB
        Nominal input power-level -4.0 dBmV(fixed), Tx Timing offset 0
        Ranging backoff (Configured- Start 16, End 16)(Actual- Start 0, End 2)
        Ranging Insertion Interval (Configured 0 ms) (Actual 1280 ms)
        Tx backoff (Start 0, End 5)
        Modulation Profile Group 1
        Ingress-cancellation is disabled
        Minislot Size in number of Timebase Ticks is = 4
        Upstream channel ID: 2
C3#
```

```
C3#show controller fastethernet 0/0

Interface FastEthernet0/0
Hardware is ethernet
        tx_carrier_loss/tx_no_carrier=0
        tx_late_collision=0, tx_excess_coll=0
        tx_collision_cnt=0, tx_deferred=0
C3#
```

show debug

Shows the current debug state. The output of this command shows four tables:

1. Mac Addresses enabled for Debug:

   Lists the MAC addresses, MAC address masks, and debug verbosity levels of all cable modems that were specified by MAC address (e.g. **debug cable mac-address 00a0.7300.0000 ffff.0000.0000 verbose**, etc).

   The table is sorted by MAC address, and shows the latest verbosity level and MAC address mask associated with the MAC address. Thus, if two or more commands are entered with the same MAC address (but differing MAC address masks or verbosity levels), only the latest setting is displayed.

   Note: The list may include CM MAC addresses which are not yet online or are completely unknown to the CMTS.

   A single command may enable many cable modems for debugging using the MAC address mask, but would display only one entry in the table.

   This table is displayed in a form resembling a debug command to allow a user to cut and paste from the table to disable debugging on a cable modem with the specified MAC address/MAC address mask.

2. Primary SIDs enabled for Debug:

   Lists the Primary SIDs and debug verbosity levels of all cable modems that were specified by Primary SID (e.g. **debug cable sid 123 verbose**, etc).

   This table is displayed in a form resembling a debug command to allow a user to cut and paste from the table to disable debugging on a cable modem with the specified primary SID.

3. Debugging events/message types which are enabled:

   Lists all events or message types which are enabled for debug (e.g. **debug cable range**, etc).

   This table is displayed in a form resembling a debug command to allow a user to cut and paste from the table to disable debugging for a particular event or message type.

4. Contents of Cable Modem Database debug level:

   Lists the interface, primary SID (if assigned), MAC address, and debug verbosity level of all cable modems that the CMTS knows about. The table shows which current cable modems (i.e. cable modems known to the CMTS) are selected for debugging.

Example:

```
C3#show debug

Mac Addresses enabled for Debug:
debug cable mac-address 00a0.731e.3f84 ffff.ffff.ffff


Primary Sids enabled for Debug:


Debugging events/message types which are enabled:
debug cable dhcp-relay

Contents of Cable Modem Database debuglevel:
I/F     PrimSid   MAC address     Debug
C1/0/U0 1         00a0.731e.3f84  Terse

C3#
```

show environment

Displays the current chassis power supply information, fan status, and temperature readings.

Example:

```
C3#show environment

Front Panel Display : attached
        HW rev = 2, SW rev= 7

==Power supply status==
        PSU1 : on
        PSU2 : on

==Temperature status==
        CPU1 : 28.0 degrees
        CPU2 : 26.0 degrees
        Kanga1 : 32.0 degrees
        Kanga2 : 28.0 degrees

==Fan status==
        Fan upper limit 12
        Fan lower limit 2
        Fan 1 : rotating
        Fan 2 : rotating
        Fan 3 : rotating
        Fan 4 : rotating
        Fan 5 : rotating
        Fan 6 : rotating
```

```
==LCD status==
        Contrast = 1024
        Msg 1 =   Cadant C3
        Msg 2 =  CMTS
        Msg 3 = VER:2.0.3.12
        Msg 4 = TIME:01:51:
        Msg 5 = 25
        Msg 6 = WANIP:192.1
        Msg 7 = 68.32.163
        Msg 8 = CMS T:005 A
        Msg 9 = :005 R:005
        Msg 10 = DS:501.0Mhz
C3#
```

show interfaces

Syntax: **show interfaces [cable X/Y] | [fastethernet X/Y] | [stats]**

Displays statistics for the specified interface (or all interfaces if none is specified).

cable X/Y
Specify the cable interface.

fastethernet X/Y
Specify the fast ethernet interface.

loopback
Specify the loopback

stats
Shows interface packets and character in/out statistics.

See also: “show cable modem” on page 6-32.

Example:

```
C3#show interfaces

FastEthernet0/0 is up, line protocol is up
        Hardware is ethernet, address is 00a0.7384.0366
        Description: ETH WAN - Cadant C3 CMTS- Broadcom 5421 Rev A1
        Alias:
        Primary Internet Address 192.168.32.244/24
        Outgoing access-list is not set
        Inbound  access-list is not set
        MTU 1500 bytes, BW 100000 Kbit
        Half-duplex, 100Mb/s
        Output queue 0 drops; input queue 0 drops
                4008 packets input, 870984 bytes
                Received 368 broadcasts, 0 giants
                0 input errors, 0 CRC, 0 frame
                353 packets output, 50342 bytes
                0 output errors, 0 collisions
                0 excessive collisions
                0 late collision, 0 deferred
                0 lost/no carrier

FastEthernet0/1 is down, line protocol is down
        Hardware is ethernet, address is 00a0.7384.0380
        Description: ETH MGT - ARRIS C3 - Broadcom 5421 Rev A1
        Alias:
        Primary Internet Address not assigned
        Outgoing access-list is not set
        Inbound  access-list is not set
        MTU 1500 bytes, BW 100000 Kbit
        Unknown-duplex, 100Mb/s
        Output queue 0 drops; input queue 0 drops
                0 packets input, 0 bytes
                Received 0 broadcasts, 0 giants
                0 input errors, 0 CRC, 0 frame
                0 packets output, 0 bytes
                0 output errors, 0 collisions
                0 excessive collisions
                0 late collision, 0 deferred
                0 lost/no carrier

Cable1/0 is up, line protocol is up
        Hardware is BCM3212(B1), address is 0000.ca3f.63cf
        Description: DS 1 - Cadant C3 CMTS - Broadcom 3034 Rev A1
        Alias:
        Primary Internet Address not assigned
        Outgoing access-list is not set
        Inbound  access-list is not set
        MTU 1764 bytes, BW 30341 Kbit
        Output queue 0 drops; input queue 0 drops
                896 packets input, 48737 bytes
                Received 5 broadcasts
                0 input errors
                15930935 packets output, 852418352 bytes
                0 output errors
C3#
```

Example (**stats**):

```
C3#show interfaces stats
FastEthernet0/0
Switching path          Pkts In    Chars In   Pkts Out   Chars Out
Processor               4129       899510     4          579
Total                   4129       899510     4          579
FastEthernet0/1
Switching path          Pkts In    Chars In   Pkts Out   Chars Out
```

```
Processor            0          0          0          0
Total                0          0          0          0
Cable1/0
Switching path       Pkts In    Chars In   Pkts Out   Chars Out
Processor            0          0          0          0
Total                0          0          0          0
C3#
```

show interfaces cable…

Syntax: **show interfaces cable 1/0 [option]**

Displays detailed information about a specific cable interface. Each option is described in detail below. Specifying no option shows a summary of interface statistics.

Example:

```
C3#show interfaces cable 1/0

Cable1/0 is up, line protocol is up
        Hardware is BCM3212, address is 00a0.7384.0409
        Description: ARRIS C3 MAC - Broadcom 3212 Rev B0
        Internet Address is unknown
        MTU 1764 bytes, BW 29630 Kbit, DLY unknown,
        Output queue 0 drops; input queue 0 drops
                0 packets input, 0 bytes
                Received 0 broadcasts
                0 input errors
                5263471 packets output, 321551109 bytes
                0 output errors
```

show interfaces cable 1/0 classifiers

Syntax: **show interfaces cable 1/0 classifiers [classid] [verbose]**

Displays all packet classifiers for the cable interface, or detailed information about a single classifier.

show interfaces cable 1/0 downstream

Displays downstream statistics for the cable interface.

Example:

```
C3#show interfaces cable1/0 downstream

Cable1/0: downstream is up
        3125636 packets output, 190771028 bytes, 0 discards
        0 output errors
        0 total active devices, 0 active modems
C3#
```

show interfaces cable 1/0 modem

Syntax: **show interfaces cable 1/0 modem {sid}**

Displays the network settings for the cable modem with the specified SID. Use SID **0** to list all SIDs.

Example:

```
C3(config-if)#show interfaces cable 1/0 modem 0
SID   Priv bits  Type      State   IP address       method    MAC address
1038  0          modem     up      10.16.246.225    dhcp      0000.ca24.482b
1192  0          modem     up      10.16.246.126    dhcp      0000.ca24.4a83
1124  0          modem     up      10.16.246.189    dhcp      0000.ca24.43e7
1064  0          modem     up      10.16.246.188    dhcp      0000.ca24.4670
1042  0          modem     up      10.16.246.120    dhcp      0000.ca24.456d
8238  00         multicast unknown 230.1.2.3        static    0000.0000.0000
```

show interface cable 1/0 privacy

Syntax: **show interface cable 1/0 privacy [kek | tek]**

Displays privacy parameters.

Example:

```
C3#show interfaces cable 1/0 privacy

        Configured KEK lifetime value = 604800
        Configured TEK lifetime value = 43200
        Accept self signed certificates: yes
        Check certificate validity periods: no
        Auth Info messages received: 0
        Auth Requests received: 0
        Auth Replies sent: 0
        Auth Rejects sent: 0
        Auth Invalids sent: 0
        SA Map Requests received: 0
        SA Map Replies sent: 0
        SA Map Rejects sent: 0

C3#show interface cable 1/0 privacy kek

Configured KEK lifetime value = 604800
C3#show interface cable 1/0 privacy tek

Configured TEK lifetime value = 43200
```

show interfaces cable 1/0 qos paramset

Syntax: **show interfaces cable 1/0 qos paramset [sfid] [verbose]**

Displays QoS parameters for the cable interface, or the specified service flow ID. The **verbose** option provides a more detailed listing.

Example:

```
C3#show interfaces cable 1/0 qos paramset
Sfid  Type  Name             Dir Sched Prio MaxSusRate MaxBurst   MinRsvRate
1     Act                    US  BE    1    1000000    3044       0
1     Adm                    US  BE    1    1000000    3044       0
1     Prov                   US  BE    1    1000000    3044       0
32769 Act                    DS  UNK   0    5000000    3044       0
32769 Adm                    DS  UNK   0    5000000    3044       0
32769 Prov                   DS  UNK   0    5000000    3044       0
C3#
```

show interfaces cable 1/0 service-flow

Syntax: **show interfaces cable 1/0 service-flow [sfid] [classifiers | counters | qos] [verbose]**

Displays service flow statistics for the cable interface. The options are:

sfid

Displays statistics for the specified Service Flow ID, or all Service Flows if none is specified.

classifiers

Displays information about CfrId, Sfid, cable modem MAC address, Direction, State, Priority, Matches

counters

Displays service flow counters. Counters are Packets, Bytes, PacketDrops, Bits/Sec, Packets/Sec. The **verbose** option is not available for counters.

qos

Displays statistics for all Service Flow IDs: Sfid, Dir, CurrState, Sid, SchedType, Prio, MaxSusRate, MaxBrst, MinRsvRate, Throughput.

verbose

Displays selected statistics in more detail.

Example:

```
C3#show interfaces cable 1/0 service-flow
Sfid  Sid   Mac Address     Type    Dir     Curr   Active
                                            State  Time
```

```
1     1     0000.ca31.3ed0  prim    US     Active 1h53m
32769 N/A   0000.ca31.3ed0  prim    DS     Active 1h53m
C3#
```

show interfaces cable 1/0 sid

Syntax: **show interfaces cable 1/0 sid [connectivity | counters | sid]**

Displays Service Flow information for all SIDs or optionally for a single SID. The options are:

sid

Displays Service Flow information for the specified SID. The default is to show all configured SIDs.

counters

Displays information about Sid, PacketsReceived, FragComplete, ConcatpktReceived

connectivity

Displays information about Sid, Prim Mac Address, IP Address, Type, Age, AdminState, SchedType, Sfid

show interfaces cable 1/0 signal-quality

Syntax: **show interfaces cable 1/0 signal-quality [port]**

Displays signal quality for the specified upstream port (range **0** to **5**), or all ports if no port specified.

Example:

```
C3#show interfaces cable1/0 signal-quality

Cable1/0: Upstream 0 is up includes contention intervals: TRUE

Cable1/0: Upstream 1 is up includes contention intervals: TRUE
C3#
```

show interfaces cable 1/0 stats

Displays interface statistics.

Example:

```
C3#show interfaces cable1/0 stats

Cable1/0
Switching path          Pkts In    Chars In    Pkts Out    Chars Out
Processor               1118       60760       764         1060272851
Total                   1118       60760       764         1060272851
```

```
C3#
```

show interfaces cable 1/0 upstream

Syntax: **show interfaces cable 1/0 upstream [port]**

Displays upstream information for all ports, or the specified port.

Valid range: **0** to **5**.

Example:

```
C3#show interface cable1/0 upstream
Cable1/0: Upstream 0 is up, line protocol is up
        Description: US CH 1 - Cadant C3 CMTS - Broadcom 3138 Rev A2
        Alias: US CH 1 - Cadant C3 CMTS - Broadcom 3138 Rev A2
        Received 5 broadcasts, 0 multicasts, 1126 unicasts
        0 discards, 0 errors, 0 unknown protocol
        1131 packets input, 0 uncorrectable
        0 microreflections
        Total Modems On This Upstream Channel : 1 (1 active)

Cable1/0: Upstream 1 is up, line protocol is up
        Description: US CH 2 - Cadant C3 CMTS - Broadcom 3138 Rev A2
        Alias: US CH 2 - Cadant C3 CMTS - Broadcom 3138 Rev A2
        Received 0 broadcasts, 0 multicasts, 0 unicasts
        0 discards, 0 errors, 0 unknown protocol
        0 packets input, 0 uncorrectable
        0 microreflections
        Total Modems On This Upstream Channel : 0 (0 active)
C3#
```

show interfaces fastethernet X/Y…

Syntax: **show interfaces fastethernet X/Y [stats]**

Displays detailed information about a specific Ethernet interface. Each option is described in detail below. Specifying no option shows detailed interface statistics:

```
C3#show interfaces fastethernet0/0

FastEthernet0/0 is up, line protocol is up
        Hardware is ethernet, address is 0000.ca3f.63cd
        Description: ETH WAN - Cadant C3 CMTS - Broadcom 5421 Rev A1
        Alias:
        Primary Internet Address 10.1.12.45/25
        Outgoing access-list is not set
        Inbound  access-list is not set
        MTU 1500 bytes, BW 100000 Kbit
        Half-duplex, 100Mb/s
        Output queue 0 drops; input queue 0 drops
                23138 packets input, 6456298 bytes
                Received 10545 broadcasts, 0 giants
                10 input errors, 10 CRC, 9 frame
                3395 packets output, 296344 bytes
                0 output errors, 0 collisions
                0 excessive collisions
                0 late collision, 0 deferred
                0 lost/no carrier
C3#
```

show interfaces fastethernet X/Y stats

Displays a summary of interface statistics.

Example:

```
C3#show interfaces fastethernet0/0 stats

Fastethernet0/0
Switching path          Pkts In    Chars In   Pkts Out   Chars Out
Processor               9883       1251544    7991       537952
Total                   9883       1251544    7991       537952
C3#
```

show ip…

Syntax: **show ip [arp | cache | igmp | rip | route]**

Displays IP parameters. The following sub-commands are available only in privilege mode.

See also: “show ip arp” on page 6-10, “show ip igmp groups” on page 6-10, “show ip igmp interface” on page 6-10, “show ip rip” on page 6-11, “show ip route” on page 6-11.

show ip cache

Displays the IP routing cache.

show license

Displays a list of additional license features enabled on this CMTS.

Example:

```
C3#show license
----------------------------------------------------------------------
C3 - hostid 312 - Licensed Features

         * RIP              ARSVS01163
         * BRIDGE_GROUPS    ARSVS01164
----------------------------------------------------------------------
C3#
```

See also: “license” on page 6-17.

show logging

Displays event logging information.

```
C3#show logging

Syslog logging: disabled
Logging Throttling Control: unconstrained
DOCSIS Trap Control: 0x0

Event Reporting Control:
        Event            Local  Trap   Syslog  Local-
        Priority                               Volatile
        0(emergencies)   yes    no     no      no
        1(alerts)        yes    no     no      no
        2(critical)      yes    yes    yes     no
        3(errors)        no     yes    yes     yes
        4(warnings)      no     yes    yes     yes
        5(notifications) no     yes    yes     yes
        6(informational) no     no     no      no
        7(debugging)     no     no     no      no

Log Buffer (- bytes):
```

show mib

Syntax: **show mib ifTable**

Displays the current state of the ifTable MIB.

Example:

```
C3#show mib ifTable
index ifType ifAdminStatus LinkTraps ifAlias
1     ETH    up            enabled
2     ETH    down          enabled
3     CMAC   up            disabled
4     DS     down          enabled
5     US     down          disabled
6     US     down          disabled
11    US-CH  down          enabled
12    US-CH  down          enabled
C3#
```

show processes

Syntax: **show processes [cpu | memory]**

Displays information about running processes and CPU utilization. The options are:

(no option)
Show status for all processes, including stopped processes.

cpu

Show CPU usage over time.

memory

Show currently running processes.

Example:

```
 NAME        ENTRY       TID    PRI   STATUS      PC       SP     ERRNO  DELAY
---------- ------------ -------- --- ---------- -------- -------- ------- -----
tExcTask   excTask      89ef85d0   0 PEND       813f9320 89ef8400       0     0
tLogTask   logTask      89ef5a10   0 PEND       813f9320 89ef5848       0     0
tAutopsy   autopsy      89efe6e0   0 PEND       813f9320 89efe3e8       0     0
tShell     shell        896ee9a0   1 SUSPEND    8132beb0 896ee3d8       0     0
tPcmciad   pcmciad      89ef4180   4 PEND       813f9320 89ef3fb0       0     0
Scheduler  schedulerMai 89521c40  10 PEND       8132beb0 89521a00  3d0002     0
tNetTask   netTask      89908200  50 PEND       8132beb0 899080f0       0     0
tTimerSvr  TimerSvr     89efc3b0  90 DELAY      813d88f0 89efc2c0       0     1
tMdp1      MdpMain      89620040  95 PEND       8132beb0 8961ff08       0     0
tMdp2      MdpMain      89613120  96 PEND       8132beb0 89612fe8       0     0
tPortmapd  portmapd     896f11f0 100 PEND       8132beb0 896f0f40      16     0
tIgmp      igmpTask     8956bcd0 100 PEND       813f9320 8956bae8       0     0
FftMgr     fftMain      89524ae0 100 PEND       8132beb0 895249a8  3d0002     0
tRngMgr    RngMain      8955c300 107 PEND       813f9320 8955c120       0     0
tAuthMgr   AuthMain     89571b40 108 PEND       813f9320 89571918       0     0
tRegMgr    RegMain      8956eb50 109 PEND       813f9320 8956e928       0     0
tTek       BPIPKHTask   8955ea00 109 PEND       813f9320 8955e818       0     0
tDsxMgr    DsxMain      895bd750 110 DELAY      813d88f0 895bd638  3d0002     1
tBpi       BPIPTask     89568eb0 110 PEND       813f9320 89568cc8       0     0
tPPIf      PPIf_main    896dc220 115 PEND       813f9320 896dbe78       0     0
tUsDsMgr   channelMgtMa 8957f160 120 PEND       813f9320 8957ef30  3d0002     0
tCmMgr     CmmMain      89575240 120 PEND       813f9320 89575058       0     0
tBridge    bridge_main  89557e60 120 PEND       813f9320 89557c40       0     0
tDhcpRelay dhcpRelayMai 895b54c0 125 PEND       8132beb0 895b4f98       0     0
tNTPMib    NTPMibMain   89510eb0 128 PEND       813f9320 89510cc8       0     0
tDsxHelper DsxHelper    895e48a0 129 DELAY      813d88f0 895e47c8  3d0002     1
tDDMibs    DocsDevMIBMa 895b9cd0 129 PEND       813f9320 895b9af0       0     0
SysMgr     8103e688     896c2f70 130 PEND       813f9320 896c2c80   30065     0
tCmtsDebugLSM_CmtsDebug 89606200 130 PEND       8132beb0 89605ff8       0     0
tSnmpD     snmpd_main   89603fb0 130 PEND       8132beb0 89603c58  2b0001     0
tTimeout   activeTimeou 895e1df0 130 PEND       8132beb0 895e1d38       0     0
tPtyCli    cli_ptyOutpu 895df340 130 DELAY      813d88f0 895dee50  388002     8
tRomCli    cli_main     895da430 130 READY      813d9430 895d9420  388002     0
tEthMgr    ethMgtMain   89578280 130 PEND       813f9320 89578048       0     0
tFPD       fpd_main     8953e470 130 PEND+T     813f9320 8953e098  3d0004    14
tIdlRngMgr idleRingMgrM 8957a8b0 131 PEND       8132beb0 8957a778  3d0002     0
tLogEvt    LogEventTask 895b26c0 140 PEND       813f9320 895b24e0       0     0
tMTmrs     MiscTimersMa 8950c870 150 PEND       813f9320 8950c688       0     0
SysMgrMonit8103eb34     896becc0 161 PEND+T     813f9320 896beae8  3d0004   260
tDcacheUpd dcacheUpd    89ed10e0 250 READY      813d88f0 89ed0fb8   3006c     0
```

```
IdleTask   8103f1d8     89efb0b0 255 READY      8103f224 89efb020       0     0
C3#
```

Example (**memory** option):

```
C3#show processes memory
  NAME        ENTRY        TID     SIZE   CUR  HIGH  MARGIN
------------ ------------ -------- ----- ----- ----- ------
tExcTask     excTask      89ef85d0  7680   464   624   7056
tLogTask     logTask      89ef5a10  4688   456   552   4136
tAutopsy     autopsy      89efe6e0  7872   760   856   7016
tShell       shell        896ee9a0 39008  1480  1704  37304
tPcmciad     pcmciad      89ef4180  7680   464   616   7064
Scheduler    schedulerMai 89521c40 65216   576  1448  63768
tNetTask     netTask      89908200  9680   272  2040   7640
tTimerSvr    TimerSvr     89efc3b0  3776   240   824   2952
tMdp1        MdpMain      89620040 50880   312  1080  49800
tMdp2        MdpMain      89613120 50880   312  1080  49800
tPortmapd    portmapd     896f11f0  4688   688  1056   3632
tIgmp        igmpTask     8956bcd0  9920   488  1136   8784
FftMgr       fftMain      89524ae0  9920   312  1080   8840
tRngMgr      RngMain      8955c300  9920   480  1256   8664
tAuthMgr     AuthMain     89571b40  9920   552  1080   8840
tRegMgr      RegMain      8956eb50  9920   552  1080   8840
tTek         BPIPKHTask   8955ea00  8976   488  1136   7840
tDsxMgr      DsxMain      895bd750  9920   280  1112   8808
tBpi         BPIPTask     89568eb0 16064   488  3984  12080
tPPIf        PPIf_main    896dc220 102080   936  1416 100664
tUsDsMgr     channelMgtMa 8957f160 16064   560  5672  10392
tCmMgr       CmmMain      89575240  9920   488  1016   8904
tBridge      bridge_main  89557e60 102080   544  1072 101008
tDhcpRelay   dhcpRelayMai 895b54c0  9920  1320  1496   8424
tNTPMib      NTPMibMain   89510eb0 16064   488  1016  15048
tDsxHelper   DsxHelper    895e48a0  9920   216  1048   8872
tDDMibs      DocsDevMIBMa 895b9cd0 16064   480  3072  12992
SysMgr       0x008103e688 896c2f70 16064   752  4672  11392
tCmtsDebugLo SM_CmtsDebug 89606200  7776   520  1024   6752
tSnmpD       snmpd_main   89603fb0 101408   856  3536  97872
tTimeout     activeTimeou 895e1df0  9920   184   408   9512
tPtyCli      cli_ptyOutpu 895df340  9920  1264  2968   6952
tRomCli      cli_main     895da430 102080  4944  8720  93360
tEthMgr      ethMgtMain   89578280  9920   568  4112   5808
tFPD         fpd_main     8953e470 102080   984  2184  99896
tIdlRngMgr   idleRingMgrM 8957a8b0  7872   312  1080   6792
tLogEvt      LogEventTask 895b26c0 16064   480  1008  15056
tMTmrs       MiscTimersMa 8950c870 16064   488  1016  15048
SysMgrMonito 0x008103eb34 896becc0  7872   472  3688   4184
tDcacheUpd   dcacheUpd    89ed10e0  4688   296  1400   3288
IdleTask     0x008103f1d8 89efb0b0   688   144   512    176
INTERRUPT                           5008     0  1712   3296
```

```
C3#
```

Example (**cpu** option):

```
C3#show processes cpu
        Mgmt CPU clock speed = 600Mhz
        Mgmt CPU running at  13% utilization
        Usage over last  20 periods
        |15%|13%|15%|20%|20%|20%|15%|15%|13%|15%|
        |20%|15%|13%|15%|27%|13%|19%|15%|15%|13%|

        Avg usage over last  20 periods = 16%
        (Period  36 ticks unloaded)
C3#
```

show reload

Displays a list of scheduled reload times.

See also: “reload” on page 6-42.

show running-configuration

Displays the running configuration on the console (CLI). This command may be abbreviated to **show run**.

show snmp-server

Displays the SNMP configuration as it is specified in the running configuration.

show startup-configuration

Displays the startup configuration on the console (CLI). Note that this is not necessarily the same as the running configuration.

Appendix C contains an example showing the factory default configuration.

show tech-support

Prints a very detailed listing of C3 status for technical support purposes. This is a compilation of the following reports:

- show version
- show running-config
- show interfaces
- show controllers
- show cable modem
- show cable modulation-profile
- show interfaces cable 1/0 downstream

- show interfaces cable 1/0 upstream
- show processes
- show processes memory
- show memory
- show bridge
- show environment
- show snmp
- show users
- show terminal
- show IPC
- show file systems
- show file descriptors

Global Configuration Commands

To access this mode, enter the **configure terminal** command from privileged mode. In Global Configuration mode, the prompt is hostname(config)#.

In this mode, many normal user and privileged mode commands are not available. Return to privileged mode by typing **exit** or **Ctrl-Z** before using other commands.

end
exit
Ctrl-Z

Exits configuration mode and returns to privileged mode.

access-list

Defines and manages Access Control Lists (ACLs). Use ACLs to prevent illegal access to services provided by the C3, such as Telnet, DHCP relay, and SNMP, from external sources such as cable modems, CPEs or other connected devices. You can also use ACLs to prevent access to service via the CMTS; that is, traffic passing through the C3 can also be subjected to ACL based filtering.

You can define up to 30 ACLs; each ACL may contain up to 20 entries (ACEs). The C3 applies ACLs to all network traffic passing through the CMTS.

After defining ACLs, use the **ip access-group** command found on page 113 to associate each ACL with a specific interface or sub-interface.

See "Working with Access Control Lists" on page 8-6 for details about creating ACLs.

Standard ACL definition

Syntax: **[no] access-list {ACL-number} {permit | deny} {host *ipaddr* | any}**

A standard ACL allows or denies access to traffic to or from a particular IP address. The valid range for standard ACLs is **1** to **99**, or **1300** to **1399**.

Extended IP definitions

Syntax: **[no] access-list {ACL-number} {permit | deny} {protocol} {options}**

Extended ACLs support very precise definitions of packets. See "Filtering Traffic" on page 8-5 for more details.

The valid range for extended ACLs is **100** to **199**, or **2000** to **2699**.

alias

Syntax: **[no] alias {aliasname} {string}**

Creates an alias, which if entered as a command, executes the command *string*. The command string must be enclosed in quotes. Use **no alias** to remove an alias.

```
C3(config)#alias scm "show cable modem"
C3(config)#
```

arp

Syntax: **[no] arp {ipaddr} {macaddr} [cable 1/0[.s] [vlan] | fastethernet 0/n[.s] [vlan]]**

Creates or deletes a manual entry in the ARP table. You can optionally associate the entry with a specific sub-interface and VLAN ID.

See also: "show arp" on page 6-7.

banner

Syntax: **[no] banner {string}**

Sets the login banner for the CMTS to be the specified string. Use the **no banner** command to delete the banner completely.

boot system flash

Syntax: **boot system flash path/filename**

Boots the system from an alternate image on the Compact Flash disk.

> ***Note:*** Specify the drive letter in UPPER case:
>
> ```
> boot system flash C:/alternate_image.bin
> ```

See also: "show bootvar" on page 6-8, "reload" on page 6-42.

boot system tftp

Syntax: **boot system tftp filename ipaddr**

Boots the system from an alternate image with name *filename* on the TFTP server at the specified IP address.

See also: "show bootvar" on page 6-8, "reload" on page 6-42.

bridge

Syntax: **[no] bridge {n}**

Creates or removes a bridge group.

> ***Note:*** With a basic license, the two default bridge groups cannot be removed using the **no** form of this command. Use the **no bridge-**

group command to remove sub-interfaces from the default bridge groups.

See also: “bridge-group” on page 6-111, “bridge <n> bind” on page 6-68, “show bridge-group” on page 6-47, “encapsulation dot1q” on page 6-111.

bridge aging-time

Syntax: **[no] bridge aging-time {n}**

Sets the aging time (*n* = **0** to **1000000** seconds) for the learned entries in the Ethernet bridge or all bridge-groups.

Example:

```
C3(config)#bridge aging-time 300
C3(config)#
```

bridge <n> bind

Syntax: **[no] bridge {n} bind {fastethernet | cable} {A/B.C} {W} [native] {fastethernet | cable} {X/Y.Z} {V}**

Binds a sub-interface directly to another sub-interface using the specified VLAN tags. The bridge sends all traffic arriving at sub-interface *A/B.C* with tag *W* directly to sub-interface *X/Y.Z* and tags the traffic *V*. The parameters are:

n

The bridge group to use for this binding operation. The bridge group must have already been defined by using the **bridge** command. The interfaces specified in this command must be members of this bridge group.

W, V

The 802.1Q tag to be used for this interface. This tag should NOT be in use in the C3; do not add an encapsulation specification with this tag to the same interface as this command effectively does this.

native

This option can be used only on a cable interface. Where used, traffic will not be VLAN encoded when leaving this interface. Un-encoded traffic arriving at this interface is internally encoded with the nominated VLAN tag. This reduces the processing power required to bridge packets and hence speed up bridging.

Example:

```
bridge 1 bind cable 1/0.1 2 native fastethernet 0/0.1 42
```

All VSE encoded (with ID 2) traffic arriving at cable interface 1/0.1 is sent directly to interface fastethernet 0/0.1 via bridge group 1 and is tagged with VLAN ID = 42 before exiting on this interface. This process is symmetrical. All traffic arriving at physical interface fastethernet 0/0 with VLAN ID = 42 will be allocated to the logical interface fastethernet 0/0.1 and passed directly to interface cable 1/0.1 and will leave this interface untagged (i.e. untagged since the **native** option is specified).

See also: “bridge” on page 6-67, “bridge-group” on page 6-111, “show bridge-group” on page 6-47, “encapsulation dot1q” on page 6-111.

bridge find

Syntax: **bridge find cable-modem {macaddr}**

Locates a cable modem in the bridge table by the source MAC address.

cable filter

Syntax: **[no] cable filter**

Enables or disables filtering at the cable interface.

See also: “cable filter group” on page 6-69, “cable submgmt default filter-group” on page 6-82.

cable filter group

Syntax: **[no] cable filter group group-id index index-id [dest-ip ipaddr] | [dest-mask ipmask] | [dest-port dest-port] | [ip-proto <protocol>] | [ip-tos tos-mask tos-value] | [match-action accept | drop] | [src-ip ipaddr] | [src-mask ipmask] | [src-port src-port] | [status activate | deactivate] | [tcp-status activate | deactivate] | [tcp-flags flag-mask flag-value]**

Creates a filter specification for registered cable modems and hosts attached to registered cable modems. The parameters are:

Parameter	Values	Description
group-id	1 to 1024	
index-id	1 to 1024	
dest-port	0 to 65536	

Parameter	Values	Description
protocol	0 to 256	IP Protocol
	all	Match all protocols
	icmp	Match the ICMP protocol
	igmp	Match the IGMP protocol
	ip	IP in IP encapsulation
	tcp	Match the TCP protocol
	udp	Match the UDP protocol
tos-mask	0 to 255	
tos-value	0 to 255	
src-port	0 to 65536	IP source port number
flag-mask	0-63	
flag-value	0-63	
status		Row status for pktFilterEntry
tcp-status		Row status for tcpUdpEntry

See also: “Filtering Traffic” on page 8-5, “cable submgmt default filter-group” on page 6-82, “show cable filter” on page 6-29, “cable filter” on page 6-69.

Examples

Create a new filter using:

```
cable filter group <1-1024> index <1-1024>
```

Enter values for filter as required:

```
cable filter group <1-1024> index <1-1024> dest-ip <N.N.N.N>
cable filter group <1-1024> index <1-1024> dest-mask <N.N.N.N>
cable filter group <1-1024> index <1-1024> dest-port <0-65536>
cable filter group <1-1024> index <1-1024> ip-proto <0-256>
cable filter group <1-1024> index <1-1024> ip-tos <0x0-0xff(Mask)> <0x0-
0xff(Value)>
cable filter group <1-1024> index <1-1024> tcp-flags <0x0-0x3f(Mask)> <0x0-
0x3f(Value)>
cable filter group <1-1024> index <1-1024> src-ip <N.N.N.N>
cable filter group <1-1024> index <1-1024> src-mask <N.N.N.N>
cable filter group <1-1024> index <1-1024> src-port <0-65536>
```

Decide what to do if the filter matches:

```
cable filter group <1-1024> index <1-1024> match-action accept | drop
```

Activate the filter (or de-activate it):

```
cable filter group <1-1024> index <1-1024> status activate | deactivate
```

The following example creates filters to only allow SNMP traffic to/ from modems from defined management networks and to block all multicast based traffic to/from hosts.

```
! activate filters
cable filter
! turn on subscriber managment in the CMTS
cable submgmt
! up to 16 cpe addresses per modem can be learned
! by the CMTS
cable submgmt default max-cpe 16
! let the cmts learn the attached cpe ip addres up to the maximum (16)
cable submgmt default learnable
! filter cpe traffic based on learned cpe ip address up to the maximum (16)
cable submgmt cpe ip filtering
! activate the defaults defined here for all modems and attached cpe
cable submgmt default active

! assign default filters
! note can be overridden for a modem(as can all submgmt defaults)
! by submgmt TLV's in a modem config file
cable submgmt default filter-group cm upstream 3
cable submgmt default filter-group cm downstream 2
cable submgmt default filter-group cpe upstream 1
cable submgmt default filter-group cpe downstream 1
!
! block mcast traffic
cable filter group 1 index 1
cable filter group 1 index 1 src-ip 0.0.0.0
cable filter group 1 index 1 src-mask 0.0.0.0
cable filter group 1 index 1 dest-ip 224.0.0.0
cable filter group 1 index 1 dest-mask 240.0.0.0
cable filter group 1 index 1 ip-proto ALL
cable filter group 1 index 1 ip-tos 0x0 0x0
cable filter group 1 index 1 match-action drop
cable filter group 1 index 1 status activate
cable filter group 1 index 1 src-port all
cable filter group 1 index 1 dest-port all
cable filter group 1 index 1 tcp-flags 0x0 0x0

cable filter group 1 index 2
cable filter group 1 index 2 src-ip 0.0.0.0
cable filter group 1 index 2 src-mask 0.0.0.0
cable filter group 1 index 2 dest-ip 0.0.0.0
cable filter group 1 index 2 dest-mask 0.0.0.0
cable filter group 1 index 2 ip-proto ALL
cable filter group 1 index 2 ip-tos 0x0 0x0
cable filter group 1 index 2 match-action accept
cable filter group 1 index 2 status activate


! allow SNMP from the management system to modems
! allow UDP from 172.16.5.0/24 network to modems
! on 10.160.0.0/16 network
cable filter group 2 index 1
cable filter group 2 index 1 src-ip 172.16.5.0
cable filter group 2 index 1 src-mask 255.255.255.0
cable filter group 2 index 1 dest-ip 10.160.0.0
cable filter group 2 index 1 dest-mask 255.252.0.0
cable filter group 2 index 1 ip-proto UDP
cable filter group 2 index 1 ip-tos 0x0 0x0
cable filter group 2 index 1 match-action accept
cable filter group 2 index 1 status activate

cable filter group 2 index 3
cable filter group 2 index 3 src-ip 0.0.0.0
cable filter group 2 index 3 src-mask 0.0.0.0
cable filter group 2 index 3 dest-ip 0.0.0.0
cable filter group 2 index 3 dest-mask 0.0.0.0
cable filter group 2 index 3 ip-proto ALL
cable filter group 2 index 3 ip-tos 0x0 0x0
cable filter group 2 index 3 match-action drop
cable filter group 2 index 3 status activate

! allow SNMP from modems to the management system
! allow UDP from modems on 10.160.0.0/16 network
! to 172.16.5.0/24 network
cable filter group 3 index 1
cable filter group 3 index 1 src-ip 10.160.0.0
cable filter group 3 index 1 src-mask 255.252.0.0
cable filter group 3 index 1 dest-ip 172.16.5.0
cable filter group 3 index 1 dest-mask 255.255.255.0
cable filter group 3 index 1 ip-proto UDP
cable filter group 3 index 1 ip-tos 0x0 0x0
cable filter group 3 index 1 match-action accept
cable filter group 3 index 1 status activate

cable filter group 3 index 3
cable filter group 3 index 3 src-ip 0.0.0.0
cable filter group 3 index 3 src-mask 0.0.0.0
cable filter group 3 index 3 dest-ip 0.0.0.0
cable filter group 3 index 3 dest-mask 0.0.0.0
cable filter group 3 index 3 ip-proto ALL
cable filter group 3 index 3 ip-tos 0x0 0x0
cable filter group 3 index 3 match-action drop
cable filter group 3 index 3 status activate
```

cable frequency-band

Syntax: **[no] cable frequency-band {index} {band} {start start-freq} {stop stop-freq}**

Configures a frequency band with the given start and stop edge frequencies in Hz. The C3 assigns cable modems to a frequency group, restricting their upstream frequencies to a band within that group. The parameters are:

index
Specifies a frequency group. Valid range: **1** to **10**.

band
Specifies a frequency band within the group. Valid range: **1** to **10**.

start-freq
Start frequency, in Hz. Valid range: **1800000** to **68200000**; the start frequency must be lower than the stop frequency.

stop-freq
Stop frequency, in Hz. Valid range: **1800000** to **68200000**.

You can create multiple frequency bands by configuring several bands with the same value of *index* but different values of *band*.

Use the **no** form of this command to remove a band from a frequency group. Removing the last band from a group also removes the group.

The following example defines 6 cable frequency groups with one frequency band per group.

```
cable frequency-group 1 1 start 1800000 stop 68200000
cable frequency-group 2 1 start 1800000 stop 68200000
cable frequency-group 3 1 start 1800000 stop 68200000
cable frequency-group 4 1 start 1800000 stop 68200000
cable frequency-group 5 1 start 1800000 stop 68200000
cable frequency-group 6 1 start 1800000 stop 68200000
```

If you attempt to modify an existing frequency band, all upstream channels in the cable groups that use this band must fall within all the frequency bands assigned to the frequency-group.

See also: “show cable frequency-band” on page 6-31, “cable group frequency-index” on page 6-74.

cable group…

Syntax: **[no] cable group {id} {option}**

Manages cable groups. See the sections following for details.

cable group description

Syntax: **[no] cable group id description {str}**

Creates a textual description of this cable group that is displayed in the running configuration. Use the **no** form of this command to remove the current description. The parameters are:

id

The cable group identifier (1 to 6).

str

The cable group description.

See also: “show running-configuration” on page 6-64.

cable group frequency-index

Syntax: **cable group id frequency-index {freqIndex}**

Assigns a group of frequency bands to the given upstream group. Frequency bands assigned to a upstream group before adding upstream channels. The parameters are:

id

The cable group identifier (1 to 255).

freqIndex

Frequency index (1 to 10).

The C3 always ensures that the channels in a group are within the frequency bands assigned to the group, and that no channel overlap occurs.

See also: “cable frequency-band” on page 6-73, “show cable group” on page 6-31.

cable group load-balancing

Syntax: **[no] cable group id load-balancing {initial-numeric}**

Configures distribution of cable modems across grouped upstream channels.

Each upstream channel has a “group ID” assigned to it which is used to associate that channel with other upstream channels on the same physical cable.

Cable groups thus reflect the physical cable plant layout and specifically the reverse path combining of the plant. All upstream channels in the one cable groups should be available to a modem that can see any one of these channels.

Each cable group offers two configurations for load balancing:

none

No load balancing is performed. Modems come online using any upstream in the same group. Use **no cable group id load-balancing** to disable load balancing.

initial-numeric

The number of modems is evenly distributed across the available active channels in the same group. Modems are redirected to the most appropriate upstream during initial ranging. Once a modem comes online it will remain on the same channel until rebooted at which time it may be moved to another channel if appropriate.

See also: “cable upstream…cable upstream group-id” on page 6-139.

cable modem offline aging-time

Syntax: **cable modem offline aging-time {tt}**

Changes the offline aging time. The C3 removes cable modems from its database once they have been offline for the specified amount of time.

Specify the time in seconds, **3600** to **864000** (10 days). The default is **86400** (24 hours). A value of zero is not supported.

If the aging time is changed, the C3 restarts the aging timer for all modems currently offline.

See also: “clear cable modem” on page 6-28.

cable modulation-profile

Syntax: One of:
cable modulation-profile {p} {default_prof}
cable modulation-profile {p} {IUC} [advphy] [feclen] [maxburst] [guard_time] [modulation] [scram] [seed] [diff] [prelen] [lastcw]
cable modulation-profile {p} {IUC} [fec_t] [feclen] [maxburst] [guard_time] [modulation] [scram] [seed] [diff] [prelen] [lastcw]
no cable modulation-profile {p}

Creates or changes a modulation profile. Use the **no cable modulation-profile** command to remove the specified modulation profile.

> ***Note:*** If all modulation profiles are removed, modems using this CMTS go offline and do not come online again until you recreate modulation profiles referenced in the upstream interface specification.

p

Selects the modulation profile. Valid range: **1** to **10**.

default_prof

Specifies a modulation profile with reasonable defaults:

Code	Definition
qam	Create a default QAM16 modulation profile.
qpsk	Create a default QPSK modulation profile.
mix	Create a default QPSK/QAM mixed modulation profile.
advanced-phy	Create a default 64QAM profile with advanced PHY.

IUC

The interval usage code; may be:

IUC code	DOCSIS 1.0 and 1.1	Description
1	request	Request burst
2	reqdata	Request/data burst
3	initial	Initial ranging burst
4	station	Station keeping grant burst
5	short	Short grant burst
6	long	long grant burst
	ATDMA operation	
9	advPhyS	Advanced PHY Short data
10	advPhyL	Advanced PHY Long data
11	advPhyU	Advanced PHY Unsolicited Grant Service (UGS)

fec_t

The number of bytes which can be corrected per FEC codeword.

Range: 0 to 16.

For TDMA burst profiles: **fec_t** <= 10.

For IUCs 1 to 4: **fec_t** <= 10 if they are **tdma** or **tdmaAndAtdma**, <= 16 if they are being used on an ATDMA channel.

For IUCs 9 to 11: **fec_t** <= 16

feclen

The FEC codeword length in bytes, which may be between 1 and 255.

For all burst profiles (feclen + 2 * fec_t) <= 255

maxburst

The maximum burst size in mini-slots.

guard_time

The guard time, in symbols (0 to 255).

Modulation

The type of modulation to be used for the particular IUC—it may be **qpsk** or **qam16**. With the Advanced TDMA software option, the following additional modulation methods may be used: **qam8**, **qam32**, **qam64**.

scram

Defines whether or not the scrambler should be used (**scrambler** or **no-scrambler**).

seed

The scrambler seed in hexadecimal (0 to 7fff).

diff

Indicates whether differential encoding should be used (**diff** or **no-diff**).

prelen

Length of the preamble in bits (2 to 1024). For DOCSIS 1.x cable modems, the length must be divisible by 2 for QPSK and divisible by 4 for 16QAM.

lastcw

Indicates the FEC handling for the last codeword (**fixed** or **shortened**).

Example:

```
cable modulation-profile 1 request 0 16 2 8 qpsk scrambler 338 no-diff 64 fixed
cable modulation-profile 1 reqData 0 16 2 8 qpsk scrambler 338 no-diff 64 fixed
cable modulation-profile 1 initial 5 34 0 48 qpsk scrambler 338 no-diff 400 fixed
cable modulation-profile 1 station 5 34 0 48 qpsk scrambler 338 no-diff 384 fixed
cable modulation-profile 1 short 6 75 7 8 qpsk scrambler 338 no-diff 64 fixed
cable modulation-profile 1 long 8 220 0 8 qpsk scrambler 338 no-diff 64 fixed
```

Use the **no** form of this command with no parameters after **p** to remove a modulation profile.

Example:

```
C3(config)#show cable modulation-profile

Mod IUC     Type Preamb Diff FEC    FEC   Scrambl Max  Guard Last
Scrambl
                 length enco T      CW    Seed    B    time  CW
                             BYTES  SIZE          size size  short
1   request qpsk  64    no   0x0    0x10  0x152   0    8     no
yes
1   initial qpsk  640   no   0x5    0x22  0x152   0    48    no   yes
1   station qpsk  384   no   0x5    0x22  0x152   0    48    no   yes
1   short   qpsk  64    no   0x6    0x4b  0x152   14   8     no   yes
1   long    qpsk  64    no   0x8    0xdc  0x152   0    8     no   yes
1   advPhyS 64qam 104   no   0xc    0x4b  0x152   6    8     no   yes
1   advPhyL 64qam 104   no   0x10   0xdc  0x152   0    8     no   yes
2   request qpsk  64    no   0x0    0x10  0x152   2    8     no   yes
2   reqData qpsk  64    no   0x0    0x10  0x152   2    8     no   yes
2   initial qpsk  400   no   0x5    0x22  0x152   0    48    no   yes
2   station qpsk  384   no   0x5    0x22  0x152   0    48    no   yes
2   short   qpsk  64    no   0x6    0x4b  0x152   7    8     no   yes
2   long    qpsk  64    no   0x8    0xdc  0x152   0    8     no   yes
2   advPhyS  64qam 104   no   0xc    0x4b  0x152   6    8     no   yes
2   advPhyL  64qam 104   no   0x10   0xdc  0x152   88   8     no   yes
C3(config)#no cable modulation-profile 2
C3(config)#show cable modulation-profile
Mod IUC     Type Preamb Diff FEC    FEC   Scrambl Max  Guard Last
Scrambl
                 length enco T      CW    Seed    B    time  CW
                             BYTES  SIZE          size size  short
1   request qpsk  64    no   0x0    0x10  0x152   0    8     no   yes
1   initial qpsk  640   no   0x5    0x22  0x152   0    48    no   yes
1   station qpsk  384   no   0x5    0x22  0x152   0    48    no   yes
1   short   qpsk  64    no   0x6    0x4b  0x152   14   8     no   yes
1   long    qpsk  64    no   0x8    0xdc  0x152   0    8     no   yes
1   advPhyS 64qam 104   no   0xc    0x4b  0x152   6    8     no   yes
1   advPhyL 64qam 104   no   0x10   0xdc  0x152   0    8     no   yes
C3#
```

See “Default Modulation Profiles” on page C-10 for a listing of the default profiles.

cable service class

Syntax: **[no] cable service class {name} {option}**

Defines a DOCSIS 1.1 upstream or downstream service class.

The *name* is a character string that names the service class. Note that some devices, such as Touchstone Telephony Modems, use the service class name to find service flow parameters.

The *option* is one of the following:

activity-timeout {sec}
Activity timeout in seconds. Valid range: **0** to **65535** seconds.

admission-timeout {sec}
Admitted timeout in seconds. Valid range: **0** to **65535** seconds.

downstream
Specifies that this is a downstream service class.

grant-interval {usec}
Grant interval in microseconds. Valid range: **0** to **4294967295** µsec.

grant-jitter {usec}
Grant jitter in microseconds. Valid range: **0** to **4294967295** µsec.

grant-size {byte}
Grant size in bytes. Valid range: **0** to **65535** bytes.

grants-per-interval {grants}
Grants per interval. Valid range: **0** to **127** grants.

max-burst {bytes}
Max burst in bytes. Valid range: **1522** to **4294967295** bytes.

max-concat-burst {bytes}
Max concat burst in bytes. Valid range: **0** to **65535** bytes.

max-latency {usec}
Max latency in microseconds. Valid range: **0** to **4294967295** µsec.

max-rate {bps}
Max rate in bits per second. Valid range: **0** to **4294967295** bps.

min-packet-size {bytes}
Minimum packet size in bytes. Valid range: **0** to **65535** bytes.

min-rate {bps}
Minimum rate in bits per second. Valid range: **0** to **4294967295** bps.

poll-interval {usec}
Poll interval in microseconds. Valid range: **0** to **4294967295** µsec.

poll-jitter {usec}

Poll jitter in microseconds. Valid range: **0** to **4294967295** μsec.

priority

Priority. Valid range: **0** to **7**.

req-trans-policy {pattern}

Request transmission policy bit field. Valid range: **0x0** to **0xffffffff**.

sched-type {type}

Scheduling type; one of:

Type	Definition
UGS	Unsolicited grant
UGS-AD	Unsolicited grant with Activity Detection
best-effort	Best effort
non-real-time-polling	Non-real-time polling
real-time-polling	Real-time polling

status {option}

Set the operating status of this entry; one of **activate**, **deactivate**, or **destroy**.

tos-overwrite {mask}

AND this mask with the ToS field. Valid range: **0x1** to **0xff**.

upstream

Specifies that this is an upstream service class.

cable submgmt…

Syntax: **[no] cable submgmt [option]**

Enables or disables subscriber management.

The cable modem may receive subscriber management TLVs in its configuration file. The cable modem passes that information to the CMTS during the registration process.

The **default** options specify the default behavior of the C3 if it receives no subscriber management information during modem registration. Where such information is received during registration, that information overrides the defaults.

In this manner, a provisioning system retains control over CMTS behavior with respect to enforcing:

- Cable modem and CPE IP filters
- Maximum number of CPE per cable modem
- Fixing the CPE IP addresses allowed to be attached to the cable modem or allowing learnable IP addresses

See also: “cable submgmt default filter-group” on page 6-82, “Configuring Security” on page 8-1.

cable submgmt cpe ip filtering

Syntax: **[no] cable submgmt cpe ip filtering**

Enables or disables CPE IP filtering.

- If disabled, then CPE source IP address are not validated.
- If enabled, CPE IP addresses learned by the CMTS up to the maximum number allowed (**default max-cpe**) are used to validate received CPE traffic. The CMTS discards any CPE traffic received that does not match this list.

The docsSubMgtCpeIpTable may be populated by:

- using SNMP on the CMTS MIB
- information received during modem registration, this information in turn being provided to the modem by its configuration file.
- the CMTS learning CPE addresses

Subscriber management filters are designed so that they can be reassigned using the cable modem provisioning system; these defaults may be overridden using TLVs in a modem configuration file. If these filters are never going to be manipulated in this manner then you should consider using ACLs, a more suitable and more flexible static filtering mechanism.

cable submgmt default active

Syntax: **[no] cable submgmt default active**

Specifies that all modems and CPE devices are managed at the headend with the defined defaults.

This command establishes defaults for subscriber management. If the C3 receives subscriber management information during registration, that information overrides the defaults for this modem (and attached CPE).

cable submgmt default filter-group

Syntax: **cable submgmt default filter-group [cm | cpe] [upstream | downstream] {groupid}**

Assigns default filters. The filter groups themselves can be created via SNMP or using the **cable filter group** command.

See also: “Filtering Traffic” on page 8-5, “cable filter group” on page 6-69, “show cable filter” on page 6-29.

cable submgmt default learnable

Syntax: **[no] cable submgmt default learntable**

Enables automatic subscriber address learning (use **no cable submgmt learntable** to disable).

This command establishes defaults for subscriber management. This information can also be received from a modem during the modem registration process, overriding this default setting. The modem in turn receives this information in its configuration file.

See also: “cable submgmt cpe ip filtering” on page 6-81.

cable submgmt default max-cpe

Syntax: **cable submgmt default max-cpe {n}**

Sets the maximum number of allowable CPE devices on any modem. Valid range: **1** to **1024**.

cli logging

Syntax: **[no] cli logging [password | path dir | size maxsize]**

Controls CLI logging. The options are:

(no options)
Turns CLI logging on or off (**no cli logging**).

password
Turns password logging on or off.

path
The path in which the default log file will be stored. The filename will be “console.log,” “vty0.log,” “vty1.log,” “vty2.log.” or “vty3.log.”

size

Specifies the logging file size in Kbytes. Valid range: **1** to **50000**.

cli account

Syntax: **[no] cli account {account-name} [password pw | enable-password privpw | secret-password enpw]**

Sets the login name and passwords for access to the C3 command line. The parameters are:

account-name

Login name.

pw

Login password for this account.

privpw

Password to move into privilege mode for this account. This password is shown in clear text in the C3 configuration.

enpw

Set the encrypted password to move to privilege mode after login. This password is visible in the configuration file in encrypted format.

Use **no cli account** to delete a password.

clock summer-time date

Syntax: **clock summer-time {timezone} date {start} {end}**

Creates a specific period of summer time (daylight savings time) for the specified time zone. Use **clock summer-time recurring** to set recurring time changes.

The parameters are:

timezone

A time zone name. Use **clock timezone** to create the timezone.

start

The starting date and time. The format is: **day month year hh:mm**.

end

The ending date and time.

Example:

```
C3(config)#clock summer-time EDT date 1 4 2003 02:00 1 10 2003 02:00
```

clock summer-time recurring

Syntax: **clock summer-time {timezone} recurring [start end]**

Creates a recurring period of summer time for the specified time zone. Use **clock summer-time date** to set a specific period of summer time.

The parameters are:

timezone
A time zone name. Use **clock timezone** to create the timezone.

start
The starting date and time. The format is: **week day month hh:mm**

week can be **first**, **last**, or **1** to **4**

day is a day of the week (**sun** through **sat**, or **1** to **7**)

end
The ending date and time.

Example:

```
C3(config)#clock summer-time EDT recurring first sun apr 02:00 first sun oct
02:00
```

clock timezone

Syntax: **[no] clock timezone {name} {offset}**

Creates a time zone. Use **no clock timezone** to delete a configured timezone.

name
Any text string to describe the time zone.

offset
The offset, in hours (and optionally minutes), from UTC. Valid range: **–13** to **+13**.

default cm sub-interface

Syntax: **default cm subinterface {cable 1/0.s}**

Defines the sub-interface used for cable modem traffic until the cable modem receives an IP address from a DHCP server.

default cpe sub-interface

Syntax: **default cpe ipsubinterface {cable 1/0.s}**

Defines the sub-interface used as a source sub-interface for CPE traffic when that traffic has no VLAN tag or explicit mapping (using the **map-cpe** command).

elog

Syntax: **elog {ascii-dump | clear | off | on | size rows}**

Controls and displays the event log. The parameters are:

ascii-dump
Dump the log to the screen.

clear
Empty the log.

on
Turn on event logging.

off
Turn off event logging.

size
Set the size of the event log as the number of rows to be stored.

Example:

```
C3(config)#elog ascii-dump
Index      Event Code Count  First Time      Last Time       CM MAC Addr
1          82010100   16     JUL 08 18:33:33 JUL 08 18:33:48 --------------
2          82010200   1      JUL 08 18:33:48 JUL 08 18:33:48 0000.ca30.1288
3          82010400   1      JUL 08 18:33:48 JUL 08 18:33:48 --------------
4          82010100   7      JUL 15 16:43:16 JUL 15 16:54:26 --------------
5          82010100   16     JUN 26 15:25:54 JUN 26 15:26:09 --------------
etc...
C3(config)#
```

enable password

Syntax: **[no] enable password {string}**

This command sets the initial password to the specified *string*. To clear the password, use the **no enable password** command.

enable secret

Syntax: **[no] enable secret {string}**

Sets the privileged mode encrypted password to *string*. If this password is not set, then the enable password is required for privileged mode access. To clear this password, issue the **no enable secret** command.

The password *string* must be at least 8 characters long.

If both the enable and enable secret passwords have not been set, the C3 disables access to privileged mode using telnet. You can still enter privileged mode using a direct serial connection to the C3.

exception

Syntax: **[no] exception {auto-reboot | 3212-monitor {reboot | reset}}**

Enables automatic re-boot on crash, or when the C3 detects a problem on the cable interface. The parameters are:

auto-reboot
Specifies automatic reboot after a system crash.

3212-monitor
Specifies CMTS behavior upon detecting a problem on the downstream interface (**reboot** or **reset**).

file prompt

Syntax: **file prompt {alert | noisy | quiet}**

Instructs the C3 to prompt the user before performing certain types of file operations.

- If **noisy** is specified, the CMTS asks the user to confirm all file operations.
- If **alert** is specified, the CMTS asks the user to confirm only destructive file operations.
- If **quiet** is specified, the CMTS asks the user to confirm only **format** or **erase** commands.

help

Displays a list of available commands and a brief description of each command.

hostname

Sets the C3 host name.

ip default-gateway

Syntax: **[no] ip default-gateway {ipaddr}**

Sets the default gateway for DHCP relay and TFTP routing operations.

Use **show ip route** to verify the current default gateway.

> ***Note:*** This specification has no effect in “ip routing” mode. In IP routing mode, the running configuration contains the default gateway but the specification has no action.

See also: “ip route” on page 6-87.

ip domain-name

Syntax: **ip domain-name {string}**

Sets the domain name for the CMTS. The string is a domain name such as **example.net**.

The commands **hostname** and **ip domain-name** both change the SNMP variable “sysName.” For example, if sysName should be “cmts.example.net,” use the following commands to set it up:

```
hostname “cmts”
ip domain-name “arrisi.com”
```

The prompt displayed at the CLI is the hostname only; using the example above, the prompt would be cmts(config)#

ip route

Syntax: **[no] ip route {ipaddr subnet gateway} [dist]**

Adds a static route to the C3. The parameters are:

addr

Destination network or host IP address to be routed.

Note: In bridging mode, a **0.0.0.0** address and **0.0.0.0** mask has no effect. Use **ip default gateway** instead.

subnet

Netmask (or prefix mask) of the destination network or host IP address to be routed.

Note: In bridging mode, a **0.0.0.0** address and **0.0.0.0** mask has no effect. Use **ip default-gateway** instead.

gateway

IP address that has routing knowledge of the destination IP address.

dist

The optional administrative distance for this route. Valid range: **1** to **255**. Default: **1**.

In bridging mode, this command can be used to provide routing information for the DHCP relay function and specifically when “cable helper-address <N.N.N.N>” is used. The helper-address specified may not be on a subnet known to the Cadant C3 or known to the Cadant C3 default route (eg the DHCP server specified is behind an external router and this router is NOT connected to the management port).

Different gateways may be given for the same route with different administrative distances. The C3 uses the lowest administrative distance until the route fails, then uses the next higher administrative distance, and so on. Up to 6 static routes may be configured in this manner. The route to a connected subnet (subnet of a sub-interface) always has an administrative distance of 0, this is the first routeselected if there is any conflict with a static route.

In case of two static routes to the same subnet with equal administrative distances, the C3 uses the first provisioned route. If that route fails, then the C3 uses the next route. After a reboot, the C3 uses the first static route defined in the startup-configuration file. An example of this is shown following—refer to the 6 static routes (*) and (**) for network 15.0.0.0/24.

```
C3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - ICMP, B - BGP
       E - EGP, G - GGP, O - OSPF, ES - ES-IS, IS - IS-IS
       * - candidate default, > - primary route


Gateway of last resort is 10.250.96.1 to network 0.0.0.0


S*   0.0.0.0/0 [1/0] via 10.250.96.1, FastEthernet 0/1.0
     4.0.0.0/24 is subnetted, 1 subnet
R          4.4.4.0 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
<<<<< rip learned - default AD=120
     5.0.0.0/24 is subnetted, 1 subnets
S>       5.5.5.0 [130/0] via 10.250.96.7, FastEthernet 0/1.0
<<<< primary static with AD changed to 130
S            [130/0] via 10.250.96.8, FastEthernet 0/1.0
<<<< backup static
     7.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
R       7.0.0.0/24 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
R       7.0.0.0/8 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
R       7.7.0.0/16 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
     10.0.0.0/24 is subnetted, 4 subnets
C        10.7.8.0 is directly connected, Cable 1/0.9
<<<< directly connected to c3 (configured on sub-int AD=0)
C          10.250.96.0 is directly connected, FastEthernet 0/1.0
C          10.250.99.0 is directly connected, FastEthernet 0/0.0
C          10.250.103.0 is directly connected, bridge-group #0
     15.0.0.0/24 is subnetted, 1 subnets
S>       15.5.5.0 [1/0] via 10.7.8.10, Cable 1/0.9
<<< static with default AD=1 (*)
S                   [1/0] via 10.7.8.11, Cable 1/0.3
<<<< backup static, AD=1, second in config file (**)
S                    [1/0] via 10.7.8.110, Cable 1/0.3
<<<< backup static, AD=1, 3 in config file (**)
S                    [1/0] via 10.71.8.11, Cable 1/0.30
<<<< backup static, AD=1, 4 in config file (**)
S                     [1/0] via 10.72.8.11,  FastEthernet 0/0.5
```

```
<<<< backup static, AD=1, 5 in config file (**)
S                  [1/0] via 100.78.8.11, Cable 1/0.23
<<<< backup static, AD=1, 6 in config file (**)
     79.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R       79.79.79.0/24 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
R       79.79.79.101/32 [120/2] via 10.250.96.102, 00:00:03, FastEthernet 0/1.0
```

In bridging mode

One purpose for static routes is to provide routing information for the DHCP relay function. Specifically, when:

- using the **cable helper-address** command, and
- the specified helper address is not on a subnet known to the C3; for example, when the DHCP server specified is behind an external router and the router is not connected to the management port. The IP address specified with this command is not on a subnet known by the Cadant C3 IP stack. For example: the DHCP server specified is behind an external router and this router is NOT connected to the management port.

NOTE: This command cannot be used to add a default gateway in bridging mode. i.e. a “0.0.0.0 0.0.0.0” address and mask will have no effect in bridging mode. Use “ip default-gateway” instead.

In IP routing mode

This command adds a static route to the C3. Use the address mask **0.0.0.0 0.0.0.0** to add a route of last resort to the C3 routing table.

See also: “cable helper-address” on page 6-133, “cable dhcp-giaddr” on page 6-132, “ip route” on page 6-87, “show ip route” on page 6-11, “ip default-gateway” on page 6-86.

ip routing

Syntax: **[no] ip routing**

Turns on IP routing in the C3.

Must be executed from global configuration mode.

Starting IP routing retains configured bridge groups, sub-interfaces, VLAN IDs, and Layer 2 bindings between sub-interfaces. If pure IP routing is required, issue a **no bridge-group** command for each defined sub-interface.

The serial console reports the changed interface conditions. Changing from basic bridge operation to routing operation is shown as follows:

```
Init OK Logical i/f #0 (sbe0) changing state to ATTACH;
Logical i/f #1 (sbe1) changing state to ATTACH;
```

See also: “router rip” on page 6-100, “show ip route” on page 6-11, “Router Configuration Mode” on page 6-144.

key chain

Syntax: **[no] key chain {name}**

Enters keychain configuration mode for defining router authentication keychains. The [**no**] form of this command removes a keychain. In keychain configuration mode, the prompt is hostname(config-keychain)#. The commands shown following are valid in keychain configuration mode.

end

Exits configuration mode to privileged mode.

exit

Exits keychain configuration mode to configuration mode.

help

Displays a brief help message.

key-id

Syntax: **[no] key-id {n}**

Enters individual key configuration mode for the specified key (valid range: **0** to **255**). Upon entering the command, the prompt changes to hostname(config-keychain-key)#. Commands available are:

Command	Description
accept-lifetime *starttime* {**duration** *n* | **infinite** | *stoptime*}	Sets the accept lifetime for the key. The parameters are: *starttime*, *stoptime*: the time to start and stop accepting this key. The format is *hh*:*mm*:*ss day month year* **duration**: the number of seconds to accept this key. Valid range: **1** to **2147482646** seconds. **infinite**: always accept this key. The default is to accept the key immediately, with an infinite lifetime.
end	Exit to keychain configuration mode.
exit	Exit configuration mode to privileged mode.
help	Display this list of subcommands.

Command	Description
[no] key-string *name*	Set or delete the text for this key.
send-lifetime *start-time* {**duration** *n* | **infinite** | *stoptime*}	Sets the send lifetime for the key. The parameters are: *starttime*, *stoptime*: the time to start and stop sending this key. The format is *hh*:*mm*:*ss day month year* **duration**: the number of seconds to send this key. Valid range: **1** to **2147482646** seconds. **infinite**: always send this key. The default is to allow sending the key immediately, with an infinite lifetime.
show item	Show system info.

The [**no**] form of this command removes the specified key from the keychain.

See also: “ip rip authentication” on page 6-115.

line

Syntax: **line {console | vty start end}**

Configures default CLI parameters for the current user. When a new user logs into the CLI, the default CLI parameters come from the running-configuration line specifications. You can use the **terminal** commands to change your settings for the current session, but the settings revert to the defaults on the next login. The options are:

console
Configure the serial console.

vty <start> <end>
Configure a range of telnet sessions.

Upon entering the line command, the prompt changes to hostname(config-keychain-key)#. Commands available are:

Command	Description
end	Exit configuration mode.
exit	Exit configuration mode.
help	Display this list of subcommands.
length	Change the number of lines in the terminal window.
[no] monitor	Turn on debug output. Use the no option to turn off debug output.
show item	Show system info.

Command	Description
timeout	Set the inactivity timeout.
width	Change the number of columns in the terminal window.

Example:

```
C3(config)#line vty 0 3

Configuring telnet lines 0 to 3
C3(config-line)#timeout 0
C3(config-line)#exit

C3(config)#
```

login user

Syntax: **[no] login user [name string1] | [password string2]**

Changes the user level login name and password for vty (telnet) sessions.

See also: “Initial Configuration” on page 2-12 to set the password for privilege access level.

Example:

```
C3#login user ?
name               - Change login user name
password           - Change login user password

C3#login user name ?
<STRING>           -

C3#login user name arris
C3#login user password c3cmts
C3#
```

logging buffered

Syntax: **[no] logging buffered [severity]**

Enables local logging of events in a circular buffer. If not buffered, events are written only to the console. The option is:

severity
Severity level, 0 to 7.

logging on

Syntax: **[no] logging on**

Enables all syslog messages, traps, and local logging. To disable, use the **no logging on** command.

logging severity

Syntax: **[no] logging severity {level} {local | no-local} {trap | no-trap} {sys | no-sys} {vol | no-vol}**

Controls event generation by the severity level of the event. The parameters are:

level
Configure the specified severity level.

local or no-local
Enable or disable local logging for the specified security level.

trap or no-trap
Enable or disable trap logging for the specified security level.

sys or no-sys
Enable or disable syslog logging for the specified security level.

vol or no-vol
Enable or disable local volatile logging for the specified security level.

Factory default settings are:

- logging thresh none
- logging thresh interval 1
- logging severity 0 local no-trap no-sys no-vol
- logging severity 1 local no-trap no-sys no-vol
- logging severity 2 local trap sys no-vol
- logging severity 3 no-local trap sys vol
- logging severity 4 no-local trap sys vol
- logging severity 5 no-local trap sys vol
- logging severity 6 no-local no-trap no-sys no-vol
- logging severity 7 no-local no-trap no-sys no-vol

See also: “elog” on page 6-85, “logging thresh” on page 6-94, “logging trap” on page 6-95, “logging syslog” on page 6-94, “logging buffered” on page 6-92.

logging syslog

Syntax: **[no] logging syslog [host ipaddr | level]**

Enables syslog logging to the specified IP address, or set the syslog logging severity level (**0** to **7**).

Use the **no** form of this command to clear the syslog IP address. If no IP addresses are specified, the C3 sends no syslog messages.

logging thresh

Syntax: **logging thresh {all | at events1 | below events2 | interval sec | none}**

Limits the number of event messages generated. The parameters are:

all
: Block logging of all events.

at
: Set the numbers of events to allow. Valid range: **0** to **2147483647** events.

below
: Maintain logging below this number of events per interval. Valid range: **0** to **2147483647** events.

interval
: Set the event logging event interval (used with **below**). Valid range: **1** to **2147483647** seconds.

none
: Set the logging threshold to be unconstrained.

Factory default settings are:

- logging thresh none
- logging thresh interval 1
- logging severity 0 local no-trap no-sys no-vol
- logging severity 1 local no-trap no-sys no-vol
- logging severity 2 local trap sys no-vol
- logging severity 3 no-local trap sys vol
- logging severity 4 no-local trap sys vol
- logging severity 5 no-local trap sys vol

- logging severity 6 no-local no-trap no-sys no-vol
- logging severity 7 no-local no-trap no-sys no-vol

See also: “logging severity” on page 6-93, “logging thresh” on page 6-94, “logging trap” on page 6-95, “logging syslog” on page 6-94, “logging buffered” on page 6-92.

logging trap

Syntax: **[no] logging trap [level]**

Enables or disables transmission of SNMP traps. To disable, use the **no logging trap** command.

The optional ***level*** (0 to 7) logs all traps with a priority higher or equal to the level specified.

logging trap-control

Syntax: **[no] logging trap-control {val}**

Sets the value of the docsDevCmtsTrapControl MIB to enable or disable CMTS SNMP traps. Use a hexadecimal value for **val**. The MIB consists of 16 bits, with bit 0 being the most significant bit. Set a bit to **1** to enable the corresponding trap, **0** to disable it. The bits are:

Bit	Name	Description
0	cmtsInitRegReqFailTrap	Registration request fail
1	cmtsInitRegRspFailTrap	Registration response fail
2	cmtsInitRegAckFailTrap	Registration ACK fail
3	cmtsDynServReqFailTrap	Dynamic Service request fail
4	cmtsDynServRspFailTrap	Dynamic Service response fail
5	cmtsDynServAckFailTrap	Dynamic Service ACK fail
6	cmtsBpiInitTrap	BPI initialization
7	cmtsBPKMTrap	Baseline Privacy Key Management
8	cmtsDynamicSATrap	Dynamic Service Addition
9	cmtsDCCReqFailTrap	Dynamic Channel Change request fail
10	cmtsDCCRspFailTrap	Dynamic Channel Change response fail
11	cmtsDCCAckFailTrap	Dynamic Channel Change ACK fail

mib ifTable

Syntax: **mib ifTable {index} {down_ifAdmin | test_ifAdmin | up_ifAdmin} {disable_ifLinkTrap | enable_ifLinkTrap} {alias}**

Sets or overrides the admin state of interfaces. The parameters are:

index

The IfIndex of the interface to change:

1—the FE0 Ethernet port (fastethernet 0/0)

2—the FE1 Ethernet port (fastethernet 0/1)

3—the MAC layer cable interface

4—the downstream cable interface

5 to **10**—the upstream cable interfaces

11 to **16**—the upstream cable channels

down_ifAdmin

Sets the interface state to administratively down.

up_ifAdmin

Sets the interface state to administratively up.

test_ifAdmin

Sets the interface state to administratively test.

disable_ifLinkTrap

Do not generate traps if this interface changes state. This is the default state for interfaces of type **docsCableMaclayer** and **docsCableUpstream**.

enable_ifLinkTrap

Generate traps if this interface changes state. This is the default state for interfaces of type **ethernetCsmacd**, **docsCableDownstream**, or **docsCableUpstreamChannel**.

alias

Display this interface name.

The command “shutdown” and “no shutdown” provides a CLI means to shutdown or enable an interface but with the cable upstream and cable downstream interfaces, the interface is really composed of a CABLEMAC part and PHY part—the state of both interfaces in the MIB really define the state of the interface being referenced by the “shutdown” command.

If SNMP is used to change the state of one interface of such a “pair” and not the other interface, the CLI state of “shutdown” or “no shutdown” no longer applies—the user cannot know for sure from the CLI what is happening. Thus, the running configuration includes the current state of all interfaces and the CLI allows correction of such inconsistencies without using SNMP using the **mib** command (if the state has been altered remotely by SNMP). This possibility can occur on the downstream and upstream interfaces.

Example: what changes when an interface is shutdown in a 1x2 ARRIS Cadant C3.

```
C1000XB#conf t

C3(config)#interface cable 1/0
C3(config-if)#no cable upstream 0 shutdown
C3(config-if)#no cable upstream 1 shutdown
C3(config-if)#show run | inc MIB
MIB ifTable 1 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 2 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 3 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 4 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 5 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 6 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 11 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 12 up_ifAdmin Enable_ifLinkTrap " "

Or from an SNMP viewpoint:

SNMP table , part 2

index                                           Descr
    1 ETH WAN - ARRIS C3 - Broadcom 5421 Rev A1
    2 ETH MGT - ARRIS C3 - Broadcom 5421 Rev A1
    3     MAC - ARRIS C3 - Broadcom 3212 Rev B1
    4    DS 1 - ARRIS C3 - Broadcom 3034 Rev A1
    5 US IF 1 - ARRIS C3 - Broadcom 3138 Rev A2
    6 US IF 2 - ARRIS C3 - Broadcom 3138 Rev A2
   11 US CH 1 - ARRIS C3 - Broadcom 3138 Rev A2
   12 US CH 2 - ARRIS C3 - Broadcom 3138 Rev A2

SNMP table , part 3

index                Type
    1      ethernetCsmacd
    2      ethernetCsmacd
    3   docsCableMaclayer
    4 docsCableDownstream
    5   docsCableUpstream
    6   docsCableUpstream
   11                 205
   12                 205

SNMP table , part 7

index AdminStatus
```

```
    1         up
    2         up
    3         up
    4         up
    5         up
    6         up
   11         up
   12         up


C3(config-if)#cable upstream 1 shutdown
C3(config-if)#show run | inc MIB
MIB ifTable 1 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 2 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 3 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 4 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 5 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 6 down_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 11 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 12 down_ifAdmin Enable_ifLinkTrap " "

SNMP table , part 7

index AdminStatus
    1         up
    2         up
    3         up
    4         up
    5         up
    6       down
   11         up
   12       down
```

Standard IANAtypes	**Description**
docsCableMaclayer(127)	CATV MAC Layer
docsCableDownstream(128)	CATV Downstream interface
docsCableUpstream(129)	CATV Upstream interface
docsCableUpstream(129)	CATV Upstream interface
docsCableUpstreamChannel(205)	CATV Upstream Channel
docsCableUpstreamChannel(205)	CATV Upstream Channel
docsCableUpstreamChannel(205)	CATV Upstream Channel
docsCableUpstreamChannel(205)	CATV Upstream Channel

Corresponding SNMP MIB variables

Parameter	MIB variable
<index>	ifIndex
downIfAdmin	ifAdminStatus
testIfAdmin	ifAdminStatus
upIfAdmin	ifAdminStatus
disable_ifLinkTrap	ifLinkUpDownTrapEnable
enable_ifLinkTrap	ifLinkUpDownTrapEnable
<alias>	ifAlias

Example: The current state of all the interfaces is reported in the running configuration.

```
C3#show run | inc MIB
MIB ifTable 1 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 2 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 3 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 4 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 5 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 6 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 11 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 12 up_ifAdmin Enable_ifLinkTrap " "
```

no community

Syntax: **no community {string}**

Automatically removes and cleans up the community entry, users, groups, and views for the specified community. It can be used instead of **no snmp-server group**. Since many communities could be linked to the same group, it is safer to use **no community** to avoid disabling other communities by accident.

See also: “snmp-server group” on page 6-103.

ntp

Syntax: **[no] ntp {server ipaddr} [interval int | delete | disable | enable | master]**

Configures C3 time and date using an external NTP server. The parameters are:

server
Sets the address of the Network Time Protocol server.

delete
Removes the specified NTP server from the list.

disable
Disables polling of the specified server.

enable
Enables polling of a previously disabled server.

interval
The time, in seconds, the C3 waits between NTP updates. Valid range: **1** to **2147483647** seconds.

master
Designates the specified server as the master.

router rip

Syntax: **[no] router rip**

Enter router configuration mode.

IP routing must be enabled and licensed before this command will be executed. If IP routing is not enabled, the CMTS generates an error message.

See also: "Router Configuration Mode" on page 6-144.

snmp-access-list

Syntax:**[no] snmp-access-list {list-name} {deny | permit} {any | host {host-name | ipaddr} [port port] | subnet mask}**

Creates an SNMP access list. The parameters are:

host-name
The FQDN of the host.

port
Port number. Valid range: **0** to **65536**.

ipaddr
The host IP address.

subnet
Subnet from which access to be controlled.

mask
Subnet mask for this subnet.

snmp-server

The **snmp-server** commands are designed around the SNMPv3 framework. Internally the C3 SNMP agent exclusively processes all SNMP transactions as SNMPv3 messages and communicates with external

SNMP entities. The SNMPv3 agent can translate incoming and outgoing SNMP messages to and from SNMPv1, SNMPv2, and SNMPv2c.

The following commands are provided in logical rather than alphabetical order to make understanding easier.

- A **view** defines what part of a MIB can be accessed.
- A **group** defines what operations can be performed on a view with a security model.
- A **user** is assigned to a group but user must have same security model.
- A **notification** security model is assigned to a user.
- A **host** is assigned to a security model to receive traps or informs.

Example shown step by step on the following command specifications:

```
C3(config)# snmp-server view MyTrapNotify internet included
C3(config)# snmp-server group MyGroup v2c notify MyTrapNotify
C3(config)# snmp-server user MyCommunity MyGroup v2c access-list Trap
C3(config)# snmp-server notif-sec-model MySecurity MyCommunity v2c security-
model v2
C3(config)# snmp-server host MyTrapReceiver MySecurity 192.168.250.107 traps
C3(config)# snmp-server enable traps
```

The host now receives traps or informs from the defined subset (internet) of the C3 MIB using defined security.

snmp-server view

Syntax: **[no] snmp-server view {view-name} {mib-family} [mask mask] {excluded | included}**

Creates or adds to an existing SNMP MIB view. A view defines which MIB sub-tree (MIB families) can be acted upon by an SNMP transaction. A transaction is defined by the **snmp-server group** command, and may be read/write or notify.

The parameters are:

view

Specifies the SNMP view by name. The factory default configuration includes two predefined views, **docsisManagerView** and **internet** (see below for details).

mib-family

Specifies a MIB sub-tree by name, and whether that sub-tree is to be included or excluded in this view.

To add other MIB families in the same view, repeat this command with the same view name and a different MIB family.

mask

A bit mask, used to create more complex rules. The mask is a list of hexadecimal octets separated by colons, such as **a0:ff**. The most significant bit of the first octet corresponds to the left-most identifier in the OID. Thus, the command **snmp-server view test 1.3.5 mask A0 excluded** matches OIDs starting with 1.1.5, but not with 1.3.4 since the first and third bits of the mask are 1s.

Views are unique and are stored in the SNMP table:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpVacmMIB.vacmMIBViews ;
.vacmViewTreeFamilyTable
```

In this SNMP table, views are indexed by the view name and the MIB subtree OID.

The factory default views are:

internet

A pre-defined view that includes all OIDs under **iso.org.dod.internet**.

default

If the C3 is rebooted with no startup-configuration, the default configuration has no SNMP settings. When a community is created with the **snmp-server community** command, the view used is called “default.”

The example shown following defines a view which includes all OIDs under iso.org.dod.internet. For a notification view, it means that only notifications whose OIDs starts with iso.org.dod.internet can be sent by a user, the user being a member of a group, a group defining actions that can be taken with this view.

Although the MIB subtree “internet” is used in the following example, the sub-tree can be specified using the SNMP interface to the C3.

```
C3(config)# snmp-server view MyTrapNotify internet included
```

The following example shows SNMP parameters created for a default view.

```
C3(config)#snmp-server community public ro
C3(config)#
C3(config)#show snmp-
snmp-server contact "support@arrisi.com"
```

```
snmp-server location "3871 Lakefield Drive, Suite 300, Suwanee, GA
30024"
snmp-server engineboots 1
snmp-server view "default" "iso" included
snmp-server view "default" "snmpResearch" excluded
snmp-server view "default" "snmpTargetMIB" excluded
snmp-server view "default" "snmpNotificationMIB" excluded
snmp-server view "default" "snmpUsmMIB" excluded
snmp-server view "default" "snmpVacmMIB" excluded
snmp-server view "default" "snmpCommunityMIB" excluded
snmp-server group "public" v1 read "default"
snmp-server group "public" v2c read "default"
snmp-server user "public" "public" v1
snmp-server user "public" "public" v2c
snmp-server community-entry "Community1" "public" "public"
C3(config)#
```

See also: “no community” on page 6-99.

snmp-server group

Syntax: **[no] snmp-server group {group-name} {v3 {auth | noauth | priv} | v2c | v1} [notify view] [read view] [write view]**

Defines one or more transaction types a user can perform: read transaction, write transaction, or notify transaction. Each enabled transaction type must reference a view (defined using **snmp-server view**).

A group is identified by a group name (*group-name*), a security model, and the referenced *view*.

In a group, you can set a **read** view, a **write** view, and a **notify** view. A read view and a write view allows a user to respectively do SNMP GET and SNMP SET transactions on some MIB families (defined by the respective views). The **notify** view supports SNMP TRAP transactions.

The C3 predefines two groups, **public** and **private**, which correspond to the public and private SNMP community strings. The **public** group has read access; the **private** group has read and write access.

The example following and the example at the top of this section is focused on notification, but you can also create extra SNMP access lists to extend the default public and private community strings. For example, to disable the default public and private community strings, use the following commands:

```
no snmp-server group public v1
no snmp-server group public v2c
no snmp-server group private v1
no snmp-server group private v2c
```

To enable them again, use the following commands:

```
snmp-server group public v1 read default
snmp-server group public v2c read default
snmp-server group private v1 read default write default
snmp-server group public v2c read default write default
```

> ***Note 1:*** "default" is a predefined view in the C3 that allows access to all MIBs under the ISO family tree. Similarly, "public" and "private" are pre-defined group names allowing read access and read/write access, respectively.
>
> ***Note 2:*** A user (created by **snmp-server user**) can only be part of a group if they share the same security model.

Groups are unique and are stored in the SNMP table vacmAccessTable and users are stored in vacmSecurityToGroupTable:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpVacmMIB.vacmMIBObjects ;
.vacmSecurityToGroupTable
```

and

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpVacmMIB.vacmMIBObjects ;
.vacmAccessTable
```

Example:

```
C3(config)# snmp-server group MyGroup v2c notify MyTrapNotify
```

To add **MyCommunity** as a community string for SNMPv2c GETs as well as for notifications, use the following command:

```
C3(config)# snmp-server MyGroup v2c read myTrapNotify notify MyTrapNotify
```

Now **MyGroup** may be used as view for both SNMP TRAP and SNMP GET transactions.

See also: "no community" on page 6-99.

snmp-server user

Syntax (v1, v2c): **[no] snmp-server user {username} {group} {v2c | v1} [snmp-access-list list]**

Syntax (v3): **[no] snmp-server user {username} {group} v3 [{auth {md5 | sha} passwd [priv des56 passwd2]} | enc] [snmp-access-list list]**

Defines an SNMP user. The parameters are:

username

Specifies the user name string.

group

Specifies the user security model group (**snmp-server group**).

v3|v2c|v1

Specifies the SNMP version (and security model) to use. This must match the SNMP version specified in the group definition.

list

defines what ranges of IP addresses can perform gets/sets or receive notifications from SNMP

A user must be part of a group, which defines what type of transactions that user may perform. Use **snmp-server group** to create groups.

The **snmp-access-list** option applies only to notifications and defines which “notifications receivers” can receive notifications from that user. This argument is optional and if it is left out then all notification listeners are notified from the user.

Valid notifications receivers are defined by a list of rows in:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpNotification ;
.snmpNotifyObjects.snmpNotifyTable.
```

Each row in this table is identified by a tag and defines the notification transport model. This table is not editable from the C3 CLI, but the C3 predefines two rows whose tags are **Trap** and **Inform** (the name implies the notification model). See “snmp-server host” on page 6-107 for more information.

Users are unique and are stored in the SNMP table:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpUsmMIB.usmMIBObjects ;
.usmUser.usmUserTable
```

> ***Note:*** SNMPv3 uses a “user” security model for transactions. A user is defined by a security name and a security model (SNMPv1, SNMPv2, SNMPv3, etc...). SNMPv1 and SNMPv2 use a community string instead of a user. Thus, the C3 automatically converts a user name to a community string when a SNMPv3 message is converted to SNMPv2 and vice-versa.

Example:

```
C3(config)# snmp-server user MyCommunity MyGroup v2c ;
 access-list Trap
```

snmp-server notif-sec-model

Syntax: **[no] snmp-server notif-sec-model {security-identifier} {user-name-string} {v1 | v2c | v3} {security-model {v1 | v2 | usm {auth | priv}}}**

Defines a notification security model entry with identifier *security-identifier* and assigns this model to user *user-name-string*.

A notification security model entry is used to define the parameters for the creation of traps and inform packets for a security model (SNMPv1, SNMPv2, SNMPv2c, SNMPv3, etc...). Those required parameters are a security model, user and one of the following authentication and privacy combinations:

- no authentication, no privacy
- need authentication, no privacy
- no authentication, need privacy
- need authentication, need privacy

The authentication and privacy schemes are selected in the user definition (SHA1, MD5, etc. for authentication and DES, etc. for privacy).

Only an SNMPv3 notification security model supports authentication and privacy schemes, hence no combination needs be specified for SNMPv1, SNMPv2, or SNMPv2c models whose schemes defaults to no authentication, no privacy. However, for these models, a community string is required, which is specified by the security name in the user definition.

The SNMP table:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpCommunityMIB
.snmpCommunityObjects.snmpCommunityTable
```

maps a security name to a community string, and using this CLI command implicitly creates an entry in this table where the security name and community string are identical.

Network security models are stored in the SNMP table:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpTargetMIB
.snmpTargetObjects. snmpTargetParamsTable
"
```

Example:

```
C3(config)# snmp-server notif-sec-model MySecurity MyCommunity v2c
security-model v2
```

snmp-server host

Syntax: **[no] snmp-server host {notification-identifier} {security-identification} {ipaddr | hostname} {traps | informs} [udp-port port [timeout time [retries retry]]]**

Defines a host for each notification target or receivers. A host definition requires a notification security model, a transport type, a host address and one or more notification transport model tags.

notification-identifier

A string identifying the notification device (the CMTS).

security-identification

The community string or password.

ipaddr

IP address of the host

hostname

Qualified name of the host

udp-port

UDP port number (default 162)

timeout

0-2147483647 seconds

retries

1 –255 retries

The CLI command defaults the transport type to UDP, hence the host address must be specified using an IP address and an optional UDP port (defaults to 162).

Notification tags are specified by the **traps** or **informs** argument, which imply the 'Trap' or 'Inform' notification transport model tag.

Hosts are stored in the SNMP table:

```
iso.dot.org.dod.internet.snmpV2.snmpModules.snmpTargetMIB
.snmpTargetObjects.snmpTargetAddrTable
```

Example:

```
C3(config)# snmp-server host MyTrapReceiver MySecurity
192.168.250.107 traps
```

More examples: set up an IP address to receive traps/informs

```
snmp-server host < notification-identifier > < security-
indentification > <N.N.N.N> traps
```

```
snmp-server host <> <> <N.N.N.N> traps udp-port <0-65535>
snmp-server host <> <> <N.N.N.N> traps udp-port <> timeout <0-
2147483647>
snmp-server host <> <> <N.N.N.N> traps udp-port <> timeout <> retries
<0-255>
snmp-server host <Notification Identifier string> <Notification
Security Identifier string> <N.N.N.N> informs
snmp-server host <> <> <N.N.N.N> informs udp-port <0-65535>
snmp-server host <> <> <N.N.N.N> informs udp-port <> timeout <0-
2147483647>
snmp-server host <> <> <N.N.N.N> informs udp-port <> timeout <>
retries <0-255>
```

snmp-server enable

Syntax: **snmp-server enable {informs | traps}**

Enables configured traps or informs.

Example:

```
C3(config)# snmp-server enable traps
```

snmp-server disable

Syntax: **snmp-server disable informs {v2c | v3}** or
snmp-server disable traps {v1 | v2c | v3}

Disables configured traps or informs.

Example:

```
C3(config)# snmp-server disable traps v2c
```

snmp-server engineid

Syntax: **snmp-server engineid remote {string} {user-name} [auth {md5 | sha}]**

Configures a remote SNMPv3 engineID. The parameters are:

string
octet string, in hexadecimal. Separated each octet by a colon.

user-name
user name as a string

md5
Use the MD5 algorithm for authorization.

sha
Use the SHA algorithm for authorization.

snmp-server community

Syntax: **[no] snmp-server community {community_name} {access} [snmp-access-list name] [view mib-family {included | excluded}]**

Allows SNMP access to the C3 from the specified IP address and subnet using the specified community name.

access

One of the following:

ro—read only

rw—read and write

snmp-access-list

Specifies a defined access list (see “snmp-access-list” on page 6-100).

view

Specifies a defined view (see “snmp-server view” on page 6-101).

Example:

```
C3(config)# snmp-access-list test permit host 1.2.3.4
C3(config)# snmp-server community jim ro snmp-access-list test
```

or

```
C3(config)# snmp-server community jim ro snmp-access-list test view
docsisManagerView included
```

snmp-server contact

Syntax: **[no] snmp-server contact {contact-string}**

Sets the contact string for the C3. Typically, the contact string contains the name and number of the person or group that administer the C3. An SNMP manager can display this information.

snmp-server location

Syntax: **[no] snmp-server location {location-string}**

Sets the system location string. Typically, the location string contains the location of the C3.

snmp-server notif-entry

Syntax: **[no] snmp-server notif-entry {name} {tag-value tag} {trap | inform}**

Configures or deletes a notification entry in the snmpNotifyTable. The parameters are:

name

The name of the notification entry. Must be a unique string, up to 32 characters long.

tag

The tag value that selects an entry in the snmpTargetAddrTable (created, for example, by the **snmp-server host** command). Use an empty string (“”) to select no entry.

trap

Messages generated for this entry are sent as traps.

inform

Messages generated for this entry are sent as informs.

snmp-server community-entry

Syntax: **[no] snmp-server community-entry {index} {community-name} {user-name}**

Configures or deletes an entry in the snmpCommunityEntry table. You can use this command to change the community entry for a user, previously defined by the **snmp-server user** command. The parameters are:

index

The name of an entry in the snmpCommunityEntry table. The **snmp-server user** command automatically creates an entry in this table.

community-name

The community name to assign to this user (defined, for example, by the **snmp-server community** command).

user-name

The user name to assign to this community entry.

Note 1: The **snmp-server user** command creates an entry with identical community and user names. If you change one or the other, the C3 looks for the community name in messages from SNMP clients.

Note 2: The user must be associated with a group of the same type (**v1** or **v2c**) for the community entry to be useful.

Interface Configuration Commands

Use Interface configuration mode to configure the cable and Ethernet interfaces. When in this mode, the prompt changes to hostname(config-if)#.

interface

Syntax: **[no] interface {type} {number}**

Enter Interface configuration mode.

no

Removes a sub-interface.

type

One of **cable** or **fastethernet**.

number

Either **X/Y** or **X/Y.Z** (defines a sub-interface).

Common Interface Subcommands

The following subcommands may be used on both cable and Ethernet interfaces.

bridge-group

Syntax: [no] bridge-group {n}

Assign this interface to the specified bridge group.

See also: "bridge" on page 6-67, "bridge <n> bind" on page 6-68, "show bridge" on page 6-47.

description

Syntax: **[no] description {text}**

Sets the textual description of the interface.

Scope: Not applicable to a cable sub-interface.

encapsulation dot1q

Syntax: **[no] encapsulation dot1q {n} [native | encrypted-multicast]**

Assigns a VLAN tag to this sub-interface. The parameters are:

native

Defines a cable-side VPN.

Only applicable to a cable interface and is used to map CPE data arriving via a modem with a matching VSE encoded VLAN tag to this interface and to the VPN supported by this sub-interface.

This VLAN tag is used internally. Outbound data is not encoded with this tag.

Note: There can be only one native VLAN specified per sub-interface.

encrypted-multicast

Downstream broadcast or multicast traffic to members of this VPN is encrypted if BPI or BPI+ is enabled. Only members of this VPN receive this multicast or broadcast.

This command is applicable on a bridged interface (no IP address) or a routed interface (has an IP address).

VLAN tags are the only way to allocate incoming fastethernet packets to a fastethernet sub-interface. This command may be omitted from only one fastethernet sub-interface per physical interface, in which case un-encoded traffic is allocated to that sub-interface. This command must be used on all other fastethernet sub-interfaces whether they are bridged or routed sub-interfaces.

The native format of this command must be used on all cable sub-interfaces made a member of a bridge group—even if VSE encoding is not going to be used.

The 802.1Q VLAN IDs specified here do not have to match the VLAN IDs used on the cable side of the C3. 802.1Q The C3 remaps VLAN IDs as required by either bridge grouping, bridge binding or routing between sub-interfaces.

See “map-cpes” on page 6-129 as all the implications for the **map-cpes** command apply to the data mapped using VSE encoding and the “native” form of this command.

See also: “bridge” on page 6-67, “bridge-group” on page 6-111, “bridge <n> bind” on page 6-68, “show bridge-group” on page 6-47, Chapter 4.

end

Exit interface configuration mode.

exit

Exit configuration mode.

help

Display help about the Interface configuration system.

interface

Syntax: **interface {cable | fastethernet | X/Y}**

Changes to a different interface configuration mode without having to exit the current configuration mode first.

See also: “interface fastethernet” on page 6-118, “interface cable” on page 6-120.

ip access-group

Syntax: **[no] ip access-group {access-list-number} {in | out}**

Associates an ACL with a specific interface.

You must assign an ACL to an interface with a direction for the ACL to have any effect. For example, only when an ACL is assigned to a CMTS interface with an **in** direction does the source IP specification refer to a device external to the CMTS.

See also: “access-list” on page 6-66, “show access-lists” on page 6-44, “Configuring Security” on page 8-1.

ip directed-broadcast

Syntax: **[no] ip directed-broadcast**

Enable or disable directed subnet broadcast forwarding on this interface.

ip l2-bg-to-bg routing

Syntax: **[no] ip l2-bg-to-bg-routing**

Enables or disables IP routing of IP packets received at a sub-interface where the sub-interface must act as an IP gateway to other C3 sub-interfaces or devices connected to other C3 sub-interfaces.

> ***Note:*** You should allow management-access on this sub-interface to allow ARP to succeed.

If a layer 2 data frame containing an IP packet arrives at a sub-interface with a layer 2 destination MAC address of the C3 sub-interface, the C3 drops the frame containing the IP packet if it is not a acceptable “management” IP packet for the C3. That is, the data frame is addressed to the C3 at layer 2 and is interpreted as C3 management traffic.

When the C3 sub-interface is being used as an IP gateway to another sub-interface, the C3 does not forward the data frame containing the IP packet to the destination device unless **ip l2-bg-to-bg-routing** is enabled. Specify the **ip l2-bg-to-bg-routing** on the sub-interface that must act as an IP gateway to allow received IP packets to be passed to the C3 IP stack. Once the IP packet has reached the IP stack, the C3 routes it to the appropriate device.

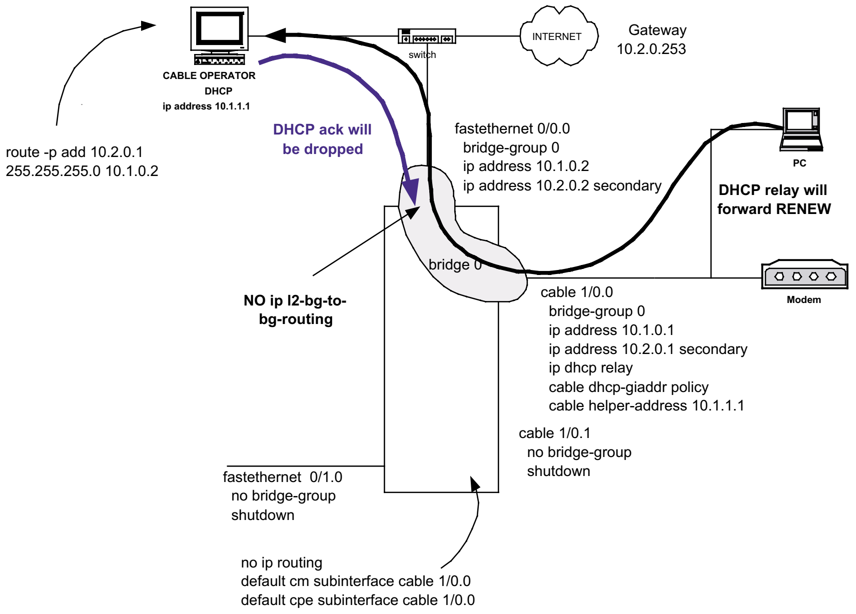
> ***Note 1:*** If the C3 is being used as an IP gateway, DHCP Renew arrives at the cable subinterface with an Ethernet MAC address of the C3 and is dropped (before seen by the DHCP Relay function) unless both **managment-access** and **ip l2-bg-to-bg-routing** are enabled on the cable sub-interface. The **management-access** command allows accepting an IP packet addressed to the C3 from this sub-interface, and **ip l2-bg-to-bg-routing** allows this IP packet to be passed to the C3 IP stack.

> ***Note 2:*** Where the C3 is not being used as the IP gateway, DHCP Relay does not need this specification to route DHCP packets, but it may be required to return an ACK to a DHCP Renew under some network conditions.

Example: DHCP renew ACK failing on one bridge group

The following example can be fixed either by:

- adding the specification **ip l2-bg-to-bg-routing** to the fastethernet 0/0.0 sub-interface
- dual homing the DHCP on the 10.2.0.0 network so that a static route is not required in the DHCP server



```
route -p add 10.2.0.1
255.255.255.0 10.1.0.2
```

```
fastethernet 0/0.0
  bridge-group 0
  ip address 10.1.0.2
  ip address 10.2.0.2 secondary
```

```
cable 1/0.0
  bridge-group 0
  ip address 10.1.0.1
  ip address 10.2.0.1 secondary
  ip dhcp relay
  cable dhcp-giaddr policy
  cable helper-address 10.1.1.1
```

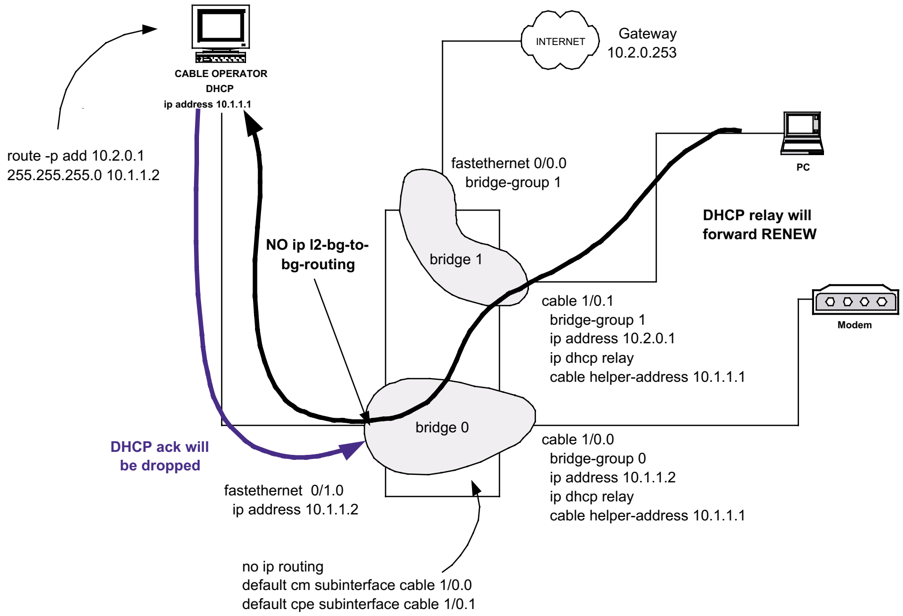
```
cable 1/0.1
  no bridge-group
  shutdown
```

```
fastethernet  0/1.0
  no bridge-group
  shutdown
```

```
no ip routing
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.0
```

Example: DHCP ACK failing across two bridge-groups

The following example can be fixed by adding the specification **ip l2-bg-to-bg-routing** to the fastethernet 0/0.0 sub-interface:



In all the above examples, the C3 DHCP relay function ensures that the RENEW is forwarded to the DHCP server, but the ACK from the DHCP server will not be addressed to any C3 IP address (addressed to the CPE) and so will not be picked up by the DHCP relay function.

ip rip authentication

Syntax: one of:

[no] ip rip authentication key-chain {name}

[no] ip rip authentication mode {text | md5}

Controls the RIP authentication method used on this interface. You can specify authentication through a key chain, using plain text passwords or MD5 passwords.

See also: “key chain” on page 6-90, “Router Configuration Mode” on page 6-144.

ip rip cost

Syntax: **ip rip cost {m}**

Manually overrides the default metric for this interface. Valid range: **1** to **16**. The default value is **1**.

ip rip default-route-metric

Syntax: **[no] ip rip default-route-metric {*m*}**

Sets the metric for default routes origniated from this interface. When 0.0.0.0/0 is advertised from a sub-interface it will have a metric set by this command. Valid range: **1** to **16**.

ip rip receive

Syntax: **[no] ip rip receive {version *versions*}**

Controls which versions of RIP packets the C3 accepts. The valid range for *versions* is 1 and 2; you can specify one or both versions with the same command.

The **no** form of this command resets the receive version on the sub-interface to the default receive version (2). To block a specific version, simply specify the alternate version. For example, to block the reception of version 2 packets, specify that only version 1 packets are to be received using the **ip rip receive version 1** command.

ip rip send

Syntax: **[no] ip rip send {version *v*}**

Controls which version of RIP packets the C3 transmits. Valid range: **1** or **2**.

The **no** form of this command resets the send version on the sub-interface to the default receive version (2). To block a specific version, simply specify the alternate version. For example, to block the sending of version 2 packets, specify that only version 1 packets are to be sent using the **ip rip send version 1** command.

ip rip v2-broadcast

Syntax: **[no] ip rip v2-broadcast**

Enables or disables broadcasting of RIPv2 updates.

ip source-verify

Syntax: **[no] ip source-verify [subif]**

Enables or disables source IP verification checks on this interface. The optional **subif** keyword verifies the IP address against the originating sub-interface subnet specifications.

This command is only valid, and has any effect only, on a routing only sub-interface.

Where a sub-interface is both a bridging and routing sub-interface—even if **ip routing** is turned on—this command has no effect as the sub-interface bridges all traffic.

ip verify-ip-address-filter

Syntax: **[no] ip verify-ip-address-filter**

Enables or disables RFC1812 IP address checks on this interface.

load-interval

Syntax: **load-interval {*time*}**

Sets the time, in seconds, to use as an interval for load averaging on this interface. Valid range: **30** to **600** seconds.

management access

Syntax: **[no] management access**

If specified for an interface, this command blocks all telnet or SNMP access through this interface.

If specified in “ip routing” mode, ARP, ICMP replies and DHCP is still allowed so that modems can acquire to a cable interface even if “no management-access” is specified.

If specified on an interface (including sub-interfaces) will block routing to this interface across bridge-group boundaries that would otherwise be possible.

CAUTION
Loss of access possible
If you use the **no** form of this command on the interface being used for management, the CMTS blocks subsequent management access.

The serial port always allows management access.

See also: “access-list” on page 6-66.

show

Syntax: **show {*item*}**

Displays parameters for the specified item.

shutdown

Syntax: **[no] shutdown**

Disables the interface. The **no** form enables the interface.

snmp trap link-status

Enable link traps.

interface fastethernet

Syntax: **interface fastethernet {0/y[.s]}**

Enters configuration mode for the specified FastEthernet interface. The valid interface numbers are:

- WAN port = **0/0**
- MGMT port = **0/1**

Example:

```
C3>enable

Password:

C3#configure terminal

C3(config)#interface fastethernet 0/0
C3(config-if)#
```

For fastethernet interfaces, the following commands are available:

duplex

Syntax: **duplex {auto | full | half}**

Sets the duplex mode of the interface. The default is **auto**, which sets both duplex mode and interface speed. It should be acceptable under most conditions.

ip address

Syntax: **ip address {ipaddr ipmask} [secondary]**

Sets the interface IP address and subnet mask. If the **secondary** option is specified, specifies a secondary IP address for the interface.

The C3 must be re-booted after changing the IP address configuration.

> ***Note:*** You can only set the management Ethernet interface primary IP address using the boot configuration. If you use the **ip address** command on the management Ethernet interface, it causes a non-fatal error and the change does not occur.

ip broadcast-address

Syntax: **ip broadcast-address {ipaddr}**

Sets the broadcast address for this interface.

ip igmp-proxy

Syntax: **[no] ip igmp-proxy [non-proxy-multicasts]**

Enables or disables IGMPv2 proxy operation on this sub-interface. For a fastethernet sub-interface to be proxy enabled, the sub-interface must:

- have an IP address configured, or
- be a member of a bridge group with an IP address configured on at least one sub-interface of the group

Each fastethernet sub-interface must be separately enabled in this manner as each sub-interface connects to a physically different network.

For example:

- if the fastethernet sub-interface is layer 2 (bridge group member) and has no IP address, then at least one sub-interface in the same bridge group must have an IP address for proxy to be enabled on that sub-interface. All cable sub-interfaces in that bridge group then operate in active mode.
- if the fastethernet sub-interface is layer 3 (routed) then all routed cable sub-interfaces operate in active mode.

In other words, if a fastethernet sub-interface is configured with an IP address, and is within a bridge group, then all cable sub-interfaces within that bridge group operate in active mode instead.

Specifying the **ip igmp-proxy** command automatically enables active IGMP routing mode on connected cable sub-interfaces. Use the **ip igmp enable** command on a per cable sub-interface basis to enable IGMP processing.

In passive mode, cable group membership information is passed to the next upstream IGMP router using the connected fastethernet sub-interfaces within the same bridge group.

When processing IGMP messages, the cable interface tracks multicast group membership in a local IGMP database and does not pass downstream a multicast stream that has no subscribing hosts (CPE or modem).

Proxy aware cable sub-interfaces also generate regular query messages downstream, interrogating multicast group membership from downstream IGMP hosts and possibly other downstream IGMP routers.

See also: “ip igmp” on page 6-125.

mac-address (read-only)

Syntax: **mac-address {aaaa.bbbb.cccc}**

Shows the MAC address of the interface.

Shown in the system configuration as a comment for information purposes only.

speed

Syntax: **speed {10 | 100 | 1000}**

Sets the speed of the interface, in Mbps. The **duplex auto** command automatically sets the interface speed as well as the duplex mode.

Scope: Not applicable to a fastethernet sub-interface.

interface cable

Syntax: **interface cable 1/0[.*s*]**

Enters configuration mode for the cable interface. The only valid entry for a cable interface is **cable 1/0**.

Example:

```
C1000XB>enable

Password:

C3#configure terminal

C3(config)#interface cable 1/0
C3(config-if)#
```

For cable interfaces, the following commands are available. Some commands are not applicable to a sub-interface where noted.

cable…

Cable interface commands are grouped as follows:

- “Cable commands (general)” on page 6-121
- “Cable commands (DHCP)” on page 6-132
- “cable downstream…” on page 6-134
- “cable upstream…” on page 6-137

Cable commands (general)

cable dci-upstream-disable

Syntax: **cable dci-upstream-disable {macaddr} {enable | disable period n}**

Instructs the addressed modem to immediately enable its upstream transmitter, or to disable it for the stated period. The parameters are:

macaddr

The MAC address of the modem.

enable

Instructs the addressed modem to enable its upstream transmitter.

disable

Instructs the addressed modem to immediately disable its upstream transmitter, no matter what state the modem is currently in.

Note: This state is not cleared in the C3 if the modem is rebooted. If the C3 is rebooted, it loses memory of this state but the modem is still disabled. The modem upstream must be re-enabled from the C3.

n

The length of time to disable the transmitter. Valid range: **1** to **4294967294** milliseconds. Use **0** to disable the modem indefinitely, and **42949672945** to enable the modem.

cable encrypt

Syntax: **cable encrypt shared-secret [string]**

Activates MD5 authentication on DOCSIS configuration files. The expected shared secret is *string*. To disable MD5 authentication, use the **no cable shared-secret** command. Use **cable encrypt shared-secret** with no string specified to enable MD5 authentication and set the expected shared secret to “DOCSIS.”

cable flap-list

Syntax: **[no] cable flap-list {aging | insertion-time | miss-threshold | size} {default | value}**

Sets parameters for the flap list. The parameters are:

aging

Sets the time that entries remain in the flap list. Use **no cable flap-list aging** to disable entry aging. Valid range: **300** to

864000 seconds (that is, 5 minutes to 10 days). Default: **259200** seconds (72 hours).

insertion-time

Sets the re-insertion threshold time. Use **no cable flap-list insertion-time** to disable re-insertion. Valid range: **60** to **86400** seconds (1 minute to 1 day). Default: **180** seconds.

miss-threshold

Sets the miss threshold. Use **no cable flap-list miss-threshold** to disable. Valid range: **1** to **12**. Default: **6**.

size

Sets the maximum number of flap list entries. Use no cable flap-list size to allow an unlimited number of entries. Valid range: **1** to **6000** entries. Default: **500**.

cable insertion-interval

Syntax: **cable insertion-interval {automatic | t}**

Sets the insertion interval. The options are:

automatic

Sets the interval based on the number of modems detected to be ranging at any particular time.

The insertion interval varies between 8 centi-seconds and 128 centi-seconds depending on whether previous opportunities were unused, used or collided. The algorithm targets a maximum interval when no modems are using the opportunities. If a collision occurs, the interval halves. If there are several unused opportunities in a row, the interval doubles. Thus, many opportunities are given when collisions occur due to many modems booting up together. Once all modems are online, the interval is set to 128 to conserve bandwidth.

When using automatic insertion intervals, set the ranging backoffs to **16**,**16**.

t

The fixed period between initial ranging opportunities, in centisecond (1/100th second) intervals.

cable map-advance

Syntax: **cable map-advance {dynamic [length] | static [length]}**

Modifies the plant length for each upstream channel when invoked with a length parameter. If a length is present, the presence of dynamic and/or static is ignored. When the length is not present, the parameters are:

dynamic
Dynamic based on current propagation time. If you specify the optional length, the C3 bases the look-ahead time on the plant length. Valid range: **0** to **161** km.

static
Static based on worst-case propagation time. If you specify the optional length, the C3 bases the look-ahead time on the plant length. Valid range: **0** to **161** km.

See also: “cable upstream plant-length” on page 6-141.

cable max-ranging-attempts

Syntax: **cable max-ranging-attempts {k}**

Sets the maximum number of ranging attempts allowed for modems. If modems exceed this limit, they are sent a ranging response with status ABORT and should proceed to attempt ranging on another advertised (via downstream UCDs) upstream channel.

Scope: Not applicable to a cable sub-interface.

Valid range: **0** to **1024**.

cable privacy

Syntax: **[no] cable privacy {option}**

Configures Baseline Privacy for the cable modems on this interface. The options are:

accept-self-signed-certificate
Allow self-signed cable modem certificates for BPI.

check-cert-validity-periods
Check certificate validity periods against the current time of day.

kek life-time n
Sets the lifetime of the Key Encryption Key (KEK).

Valid range: **0** to **6048000** seconds.

tek life-time n
Sets the lifetime of the Traffic Encryption Key (TEK).

Valid range: **0** to **6048000** seconds.

cable shared-secret

Syntax: **[no] cable shared-secret [string] [encrypted]**

Sets the shared secret to the specified *string*. If no string was specified, clear the string. This also enables or disables the CMTS MIC calculation. The **encrypted** keyword specifies that the string is to be encrypted.

The Message Integrity Check is performed during modem registration. The modem passes to the CMTS a secret given it by its configuration file and hence sourced from the provisioning systems. If this feature is turned on and the secret received in the configuration file does not match this configured value, the modem is not allowed to register.

> ***Note:*** The string is stored in the configuration in clear text. Use **cable encrypt shared-secret** if a hashed value is to be stored in the configuration.

See also: “cable encrypt” on page 6-121.

cable sid-verify

Syntax: **[no] cable sid-verify**

Enables accepting DHCP packets whose SID is zero. Use the **no** form of this command to accept such packets. The factory default settings reject DHCP packets with a SID of zero, in accordance with DOCSIS specifications. Some cable modems send these illegal packets; if your system needs to support such modems then you need to disable verification.

cable sync-interval

Syntax: **cable sync-interval {k}**

Sets the interval, in milliseconds, between SYNC messages. Valid range: **1** to **200**.

For fastest acquisition of modems, use a low number (about **20**). Sync messages use a very minor amount of downstream bandwidth.

Scope: Not applicable to a cable sub-interface.

cable ucd-interval

Syntax: **cable ucd-interval {k}**

Sets the interval, in milliseconds, between UCD messages. Valid range: **1** to **2000**.

Factory default is **2000**.

Modems check the change count in each UCD received against the last known change count. Only if this change count is different does the modem open the full UCD message and take action. If the upstream configuration is static, then decreasing this time interval achieves very little. If the upstream is being dynamically changed to move upstreams around noise, or upstream parameters are being changed rapidly for any other reason, then this time interval can be decreased.

Scope: Not applicable to a cable sub-interface.

cable utilization-interval

Syntax: **cable utilization-interval {time}**

Sets the utilization monitoring interval for US/DS channels.

Specify the time in seconds. Valid range: **0** to **86400** seconds.

ip igmp

Syntax: **ip igmp {enable | disable}**

Enable or disable active IGMP message processing on cable sub-interface, whether the processing is in active or passive mode depending on whether the cable sub-interface can "see" a proxy fastethernet subinterface.

Use this command to start IGMP query messages downstream.

Scope: Cable sub-interface only

Note that **ip igmp-proxy** must already be specified on a fastethernet interface and this fastethernet interface must be either:

- A fastethernet sub-interface with an IP address (i.e. a routed or Layer 3 sub-interface) or;
- A fastethernet sub-interface in the same bridge group as at least one other sub-interface having an IP address

See also: "ip igmp-proxy" on page 6-119.

ip igmp last-member-query-interval

Syntax: **ip igmp last-member-query-interval {val}**

Sets the interval between IGMP group specific query messages sent via the downstream to hosts.

Scope: Cable sub-interface only.

Note that **ip igmp-proxy** must already be specified on a fastethernet interface and this fastethernet interface must be either:

- A routed fastethernet sub-interface or;
- A fastethernet sub-interface in the same bridge group as at least one other routed sub-interface (a sub-interface having an IP address)

See also: “ip igmp” on page 6-125, “ip igmp-proxy” on page 6-119.

ip igmp query-interval

Syntax: **ip igmp query interval {val}**

Sets the interval between host specific query messages.

Scope: Cable sub-interface only.

Note that **ip igmp-proxy** must already be specified on a fastethernet interface and this fastethernet interface must be either:

- A routed fastethernet sub-interface or;
- A fastethernet sub-interface in the same bridge group as at least one other routed sub-interface (a sub-interface having an IP address)

See also: “ip igmp” on page 6-125, “ip igmp-proxy” on page 6-119.

ip igmp query-max-response-timeout

Syntax: **ip igmp query-max-response-timeout {val}**

Sets the maximum interval, in 1/10 second increments, the C3 waits for a response to an IGMP query. Valid range: **10** to **255**.

Scope: Cable sub-interface only.

Note that **ip igmp-proxy** must already be specified on a fastethernet interface and this fastethernet interface must be either:

- A routed fastethernet sub-interface or;
- A fastethernet sub-interface in the same bridge group as at least one other routed sub-interface (a sub-interface having an IP address)

See also: “ip igmp” on page 6-125, “ip igmp-proxy” on page 6-119.

ip igmp robustness

Syntax: **ip igmp robustness {val}**

Variable for tuning the expected packet loss on a subnet. Valid range: **1** to **255**.

Scope: Cable sub-interface only.

Note that **ip igmp-proxy** must already be specified on a fastethernet interface and this fastethernet interface must be either:

- A routed fastethernet sub-interface or;
- A fastethernet sub-interface in the same bridge group as at least one other routed sub-interface (a sub-interface having an IP address)

See also: “ip igmp” on page 6-125, “ip igmp-proxy” on page 6-119.

ip igmp verify ip-router-alert-option

Syntax: **[no] ip igmp verify ip-router-alert-option**

Enables or disables checking of the IP Router Alert option in IGMP v2 reports and leaves.

ip igmp version

Syntax: **ip igmp version {*val*}**

The version of IGMP running on the sub-interface. The value of *val* must be **2**.

Scope: Cable sub-interface only.

Note that **ip igmp-proxy** must already be specified on a fastethernet interface and this fastethernet interface must be either:

- A layer 3 fastethernet sub-interface or;
- A fastethernet sub-interface in the same bridge group as at least one other sub-interface having an IP address

See also: “ip igmp-proxy” on page 6-119.

ip-broadcast-echo

Syntax: **[no] ip-broadcast-echo**

Controls whether IP or ARP broadcasts received on the cable interface are broadcast back downstream. This may be specified per cable sub-interface.

ip-multicast-echo

Syntax: **[no] ip-multicast-echo**

Controls whether multicasts received on the cable interface are broadcast back downstream. This may be specified per cable sub-interface.

Note that the **[no]** form of this command has implications in IGMP message processing as IGMP messages from hosts are not sent back downstream.

encapsulation dot1q

Syntax: **[no] encapsulation dot1q {*n*} [native]**

Specifies the VLAN ID and encapsulation type for data leaving this interface (if **native** not specified) and the type of encapsulation and VLAN ID for data that is accepted by this interface.

native

Only applicable to a cable interface.

VLAN tag is used internally. Outbound data is not encoded with this tag.

Any un-encoded inbound data will be issued with this VLAN tag for internal use (tag will not leave the ARRIS Cadant C3)

There can be only ONE VLAN specified per sub-interfaceusing this command. Bridge bind must be used if additional encapsulation is required.

This command is applicable on a bridged interface (no IP address) or a routed interface (has an IP address).

VLAN tags are the only way to allocate incoming fastethernet packets to a fastethernet sub-interface. This command may be omitted from only one fastethernet sub-interface per physical interface in which case un encoded traffic will be allocated to this one sub-interface. This command must be used on all other fastethernet sub-interfaces whether they are bridged or routed sub-interfaces.

The native format of this command must be used on all cable sub-interfaces made a member of a bridge group—even if VSE encoding is not going to be used.

The VLAN IDs specified here do *not* have to match the VLAN IDs used on the cable side of the C3. VLAN IDs are re-mapped as required by either bridge grouping, bridge binding or routing.

See also: “bridge” on page 6-67, “bridge-group” on page 6-111, “bridge <n> bind” on page 6-68, “show bridge-group” on page 6-47, “map-cpes” on page 6-129, Chapter 5.

l2-broadcast-echo

Syntax: **[no] l2-broadcast-echo**

Enables echoing of layer 2 broadcast packets to the downstream. Use the **no** form of this command to disable broadcast echo.

l2-multicast-echo

Syntax: **[no] l2-multicast-echo**

Enables echoing of layer 2 multicast packets to the downstream. Use the **no** form of this command to disable multicast echo.

map-cpes

Syntax: **[no] map-cpes {cable 1/0.s}**

Maps all CPE attached to a modem to the specified cable sub-interface.

This command provides a static (CMTS configured) means to allocate incoming CPE packets to a defined sub-interface based on modem IP address. Use of this command implies modems are allocated to multiple subnets if more than one CPE subnet is required as there needs to be a one to one match of modem to CPE sub-interfaces.

The specified cable sub-interface may or may not have an assigned IP address.

If the specified cable sub-interface has an IP address and dhcp relay parameters are configured for this cable sub-interface, this IP address will be the giaddr address for any relayed CPE DHCP. Thus, a simple non-DOCIS aware or “standard” DHCP server can be used that allocates IP address based on the incoming DHCP giaddr value.

If the specified sub-interface does not have an IP address, it is assumed that layer 2 traffic is being bridged and that the sub-interface is a member of a bridge group.

> ***Note:*** You must specify **encapsulation dot1q <n> native** on such a sub-interface, even though VSE encoding is not being used for the sub-interface. The VLAN specification is used internally by the C3 and also allows the use of the **bridge bind** command to bind this sub-interface directly to a VLAN tagging fastethernet sub-interface if required.

If the CPE IP address must be configured on a dynamic basis or is not bound to the modem IP address—as would be the case if all modems are required to be allocated an IP address from one large single address pool—consider using VSE encoding (Chapter 8) instead of using the **map-cpes** command. VSE encoding and the use of the **encapsulation dot1q <n> native** command allows CPE attached to a modem to be allocated to a cable sub-interface based on modem configuration file specified (and hence provisioning system specified) parameters and is independent of the assigned modem IP address.

Example: One modem subnet—one CPE subnet—IP routing

```
ip routing
!
interface cable 1/0
!
ip address 10.1.0.1 255.255.0.0
ip dhcp relay
cable helper-address 10.2.0.1
cable dhcp-giaddr primary
! option 82 not really required for standard DHCP server
no ip dhcp relay information option
map-cpes cable 1/0.1
!
interface cable 1/0.1
! for CPE devices
ip address 10.11.0.1 255.255.0.0
ip dhcp relay
cable helper-address 10.2.0.1
cable dhcp-giaddr primary
! option 82 not really required for standard DHCP server
no ip dhcp relay information option
!
```

Example: One modem subnet—CPE data bridged—no IP routing

```
no ip routing
!
conf t
bridge 2
!
interface cable 1/0
!
ip address 10.1.0.1 255.255.0.0
ip dhcp relay
cable helper-address 10.2.0.1
cable dhcp-giaddr primary
! option 82 not really required for standard DHCP server
no ip dhcp relay information option
! map PPPoE CPE to another interface
map-cpes cable 1/0.1
!
interface cable 1/0.1
! for CPE devices running layer 2
! e.g. PPPoE
bridge-group 2
```

```
! add vlan spec for internal use
encapsulation dot1q 9 native
!
exit
exit
```

Example: Multiple modem subnets with *mapped* CPE subnets

```
ip routing
!
interface cable 1/0
! used for modem DHCP
ip address 10.1.0.1 255.255.0.0
ip dhcp relay
cable helper-address 10.2.0.1
cable dhcp-giaddr primary
! option 82 not really required for standard DHCP server
no ip dhcp relay information option
!
interface cable 1/0.1
! used for modem
ip address 10.10.0.1 255.255.0.0
! dhcp renews will be routed so no relay required
no ip dhcp relay
map-cpes cable 1/0.11
!
interface cable 1/0.2
! used for modem
ip address 10.20.0.1 255.255.0.0
! dhcp renews will be routed so no relay required
no ip dhcp relay
map-cpes cable 1/0.12
!
interface cable 1/0.11
! for CPE devices
ip address 10.11.0.1 255.255.0.0
! dhcp spec required for cpe dhcp
ip dhcp relay
cable helper-address 10.2.0.1
cable dhcp-giaddr primary
! option 82 not required or used by standard DHCP server
no ip dhcp relay information option
!
interface cable 1/0.12
! for CPE devices
ip address 10.12.0.1 255.255.0.0
! dhcp spec required for cpe dhcp
ip dhcp relay
cable helper-address 10.2.0.1
cable dhcp-giaddr primary
```

```
! option 82 not required or used by standard DHCP server
no ip dhcp relay information option
```

Example: self mapping using **map-cpes**

This example shows the **map-cpes** command referencing the same sub-interface. Only subnets in the mapped sub-interface are valid for CPE and so the primary sub-interface specification is also a valid subnet for CPE devices.

```
ip routing
!
interface cable 1/0.0
! valid subnet for CM and CPE devices
ip address 10.1.0.1 255.255.0.0
! valid subnets for CPE devices
ip address 10.11.0.1 255.255.0.0 secondary
ip address 10.21.0.1 255.255.0.0 secondary
ip address 10.31.0.1 255.255.0.0 secondary
ip dhcp relay
! use primary address for modem giaddr
! use first secondary address for cpe giaddr
cable dhcp-giaddr policy
! us the one dhcp server for cm and cpe
cable helper-address 10.2.0.1
! allow the dhcp server to tell what is cm what is cpe
ip dhcp relay information option
! map all cpe attached to cm using this interface
! to this interface
map-cpes cable 1/0.0
```

See also: “encapsulation dot1q” on page 6-111.

Cable commands (DHCP)

cable dhcp-giaddr

Syntax: **[no] cable dhcp-giaddr {policy | primary}**

Replaces the giaddr field in DHCP packets. The parameters are:

primary

Replaces the giaddr with the relaying interface primary IP address for cable modems and hosts.

policy

For cable modems: replaces the giaddr with the relaying interface primary IP address.

For hosts: replaces the giaddr with the relaying interface’s first secondary IP address.

If no **cable helper-address** is active, the CMTS broadcasts DHCP messages through all active Ethernet interfaces with the updated giaddr field.

See also: “ip dhcp relay” on page 6-133, “ip dhcp relay information option” on page 6-134, “cable helper-address” on page 6-133, “DHCP” on page 7-4.

cable helper-address

Syntax: **[no] cable helper-address {ipaddr} [cable-modem | host]**

Updates the giaddr field with the relaying interface primary IP address (unless **cable dhcp-giaddr policy** is active) and then unicasts the DHCP Discover or Request packet to the specified IP address.

(no options)

Unicast all cable originated DHCP broadcast messages to the specified IP address.

host

Unicast all cable originated host DHCP broadcast messages to the specified IP address.

cable-modem

Unicast all cable modem DHCP broadcast messages to the specified IP address.

You can specify up to 5 helper addresses each for cable modems and hosts (CPE), for redundancy or load sharing. The C3 performs no round-robin allocation but unicasts the relayed DHCP to each of the helper addresses specified. The cable modem or CPE responds to and interacts with the first DHCP server that replies.

See also: “ip dhcp relay” on page 6-133, “ip dhcp relay information option” on page 6-134, “cable dhcp-giaddr” on page 6-132, “Directing DHCP Broadcasts to Specific Servers” on page 7-6.

ip dhcp relay

Syntax: **[no] ip dhcp relay**

Enables the C3 to modify DHCP requests from cable modems or hosts attached to cable modems by updating the **giaddr** field with the WAN port IP address. The effect of this command is to allow the DHCP server to unicast DHCP responses back to the C3, reducing backbone broadcasts.

Use **no ip dhcp relay** (default) to disable DHCP relay. This command sends broadcast DHCP messages received at the cable sub-interface to all bridged fastethernet sub-interfaces. When specified on an IP routing-only cable sub-interface, no DHCP relay occurs at all.

See also: Chapter 7 (for details on using DHCP relay), “ip dhcp relay information option” on page 6-134, “cable dhcp-giaddr” on page 6-132, “cable helper-address” on page 6-133.

ip dhcp relay information option

Syntax: **[no] ip dhcp relay information option**

Enables modification of DHCP requests from modems or hosts attached to modems to include the modem’s address in the option 82 field. The CMTS adds option 82 information to any DHCP Discover or Request messages received from a cable modem or attached host.

DHCP relay (**ip dhcp relay**) must be active for this command to have any effect.

To disable, use **no ip dhcp relay information option** which passes relayed DHCP requests with no option 82 modification.

See also: “cable dhcp-giaddr” on page 6-132, “cable helper-address” on page 6-133, “DHCP” on page 7-4.

ip dhcp relay validate renew

Syntax: **[no] ip dhcp relay validate renew**

When this command is active, the destination IP address in a Renew message is validated against the configured helper address for cable sub-interface. If the destination address is not validated, the Renew is dropped.

See also: “cable helper-address” on page 6-133.

cable downstream…

The following downstream commands are available.

Scope: Not applicable to a cable sub-interface.

cable downstream annex

Syntax: **cable downstream annex {a | b | c}**

Sets the MPEG framing format. The format is one of:

- **A** = Europe/EuroDOCSIS
- **B** = North American DOCSIS
- **C** = Japan (6 MHz downstream, 5-65 MHz upstream)

cable downstream channel-width

Syntax: **cable downstream channel-width {6mhz | 8mhz}**

Sets the downstream channel width. Use **6Mhz** for North America and Japan, **8Mhz** for Europe.

cable downstream frequency

Syntax: **cable downstream frequency {hz}**

Sets the downstream center frequency in Hz.

Valid range: **91000000** to **857000000** for 6 MHz (North America and Japan) DOCSIS; **112000000** to **857000000** for EuroDOCSIS. The tuner has a resolution of 62500 (62.5 kHz).

> ***Note:*** If an up-converter is not installed, the CMTS disables this command.

cable downstream interleave-depth

Syntax: **cable downstream interleave-depth {I}**

Sets the FEC interleaving. Valid settings are:

Setting	R/S Interleave
128	I = 128, J = 1
64	I = 64, J = 2
32	I = 32, J = 4
16	I = 16, J = 8
8	I = 8, J = 16
12	I = 12, J = 17 (EuroDOCSIS only)

cable downstream modulation

Syntax: **cable downstream modulation {256qam | 64qam}**

Sets the downstream modulation type.

cable downstream power-level

Syntax: **cable downstream power-level {dBmV}**

Sets the downstream power level to the specified value.

Valid range: **45** to **65** dBmV.

> ***Note:*** If an up-converter is not installed, the CMTS disables this command.

cable downstream rate-limit

Syntax: **no cable downstream rate-limit** or
cable downstream rate-limit token-bucket shaping [auto-delay [auto-value val] | max-delay delay | packet-delay [packets-limit lim]]

Changes the type of rate limiting from moving average traffic shaping to “token-bucket” limiting, or to a combination of both. Use the **no** keyword with no other parameters to restore average traffic shaping. The parameters are:

shaping

Specifies the type of traffic shaping to perform.

The default is **shaping max-delay 1024**.

auto-delay

Rate shaping with automatically scaled deferral limits.

The default is **auto-value 80000**.

auto-value

The delay-bandwidth product of the rate-shaping “pipe,” in bits. For example, if the auto-value is **80000**, and the maximum bit rate is 80 kbps, the maximum delay is 1 second; if the maximum bit rate is 800 kbps, the maximum delay is 100 ms. TCP protocols (such as FTP and HTTP) require a delay-bandwidth product of at least 4 to 5 maximum-size packets (to allow a congestion window large enough to accommodate 3 duplicate ACKs for fast retransmission). In this mode, each service flow has a different maximum deferral time.

Valid range: **0** to **1000000** bits.

max-delay

The maximum deferral time of a packet. Packets which need to wait longer than this for tokens are *always* dropped. Packets which are delayed for less than one-half of this value are *not* dropped. A linear drop probability is applied between these two limits. This is a RED algorithm which is necessary for smooth TCP performance.

Valid range: **0** to **2047** milliseconds.

packet-delay

Rate shaping with packet-based deferral limits.

The default is **packets-limit 12**.

packets-limit

The maximum number of packets to defer for a given service flow. Again, RED is applied linearly between one-half this value (zero drop probability) and this value (definite drop).

Valid range: **0** to **255** packets.

The C3 limits downstream traffic to a modem based on the Class of service (DOCSIS 1.0) or Service flow specification (DOCSIS 1.1).

The C3 must enforce the CoS or QoS over a one second period. This is strictly true for DOCSIS 1.0 Class of Service; DOCSIS 1.1 Quality of Service requires the formula max(T) = T*R/8 +B to be valid for any window size T.

If the required bandwidth exceeds the enforced bandwidth, the C3 either delays the packet or (in extreme cases) drops the packet.

cable upstream...

Syntax: **cable upstream {*n*}**

Enters configuration mode for the selected upstream. Valid range: **0** to **5**.

cable upstream channel-type

Syntax: **cable upstream *n* channel-type {atdma | scdma | tdma | tdma&atdma} [modulation-profile *n*]**

Selects the desired type of channel operation.

This command also cross checks for user mis-configuration of modulation profiles and only broadcasts in the downstream applicable burst descriptor parameters and IUCs for the selected channel type.

Note: To ensure DOCSIS 1.X compatibility, specify **tdma**.

cable upstream channel-width

Syntax: **cable upstream n channel-width {w}**

Sets the upstream channel width. The channel width can be one of:

Value of W	Definition
6400000	Width 6400 KHz, Symbol rate 5120 ksym/s
3200000	Width 3200 KHz, Symbol rate 2560 ksym/s
1600000	Width 1600 KHz, Symbol rate 1280 ksym/s
800000	Width 800 KHz, Symbol rate 640 ksym/s
400000	Width 400 KHz, Symbol rate 320 ksym/s

Value of W	Definition
200000	Width 200 KHz, Symbol rate 160 ksym/s

cable upstream concatenation

Syntax: **[no] cable upstream *n* concatenation**

Enables or disables concatenation (concatenation support is on by default).

cable upstream data-backoff

Syntax: **cable upstream *n* data-backoff {automatic | *start end*}**

Set the random backoff window for data. The parameters are:

automatic
Automatically change the window.

start, end
Manually specify the window (valid range is **0** to **15**, end must be larger than start).

cable upstream description

Syntax: **[no] cable upstream *n* description {*string*}**

Sets the textual description of this upstream to *string*.

cable upstream differential-encoding

Syntax: **[no] cable upstream *n* differential-encoding**

Enable differential encoding. Use the **no** form to turn off.

cable upstream fec

Syntax: **[no] cable upstream *n* fec**

Enable Forward Error Correction (FEC). Use the **no** form to turn FEC off.

cable upstream fragmentation

Syntax: **[no] cable upstream *n* fragmentation [forced-multiple-grant *nn*} | forced-piggyback *mm*]**

Configures fragmentation for the specified interface. The options are:

(no option)
Enable normal fragmentation. Use the **no** form to disable fragmentation.

forced-multiple-grant

Forced multiple grant mode where packets are broken up into *nn* size bytes and multiple grants are scheduled to transfer these smaller packets.

Use the **no** form to disable this mode.

Valid range: **0** to **1522** bytes

forced-piggyback

Forced piggy back for fragmentation. If the cable modem is instructed to fragment a packet in to size *mm* bytes, but multiple grants are not seen by the cable modem to transfer the fragments, this mode forces the cable modem to use piggybacking to transfer the fragments.

Use the **no** form to disable this mode.

Valid range: **0** to **1522** bytes

cable upstream frequency

Syntax: **cable upstream *n* frequency {*k*}**

Sets the upstream frequency in Hz. Valid range:

- North American DOCSIS: **5000000** to **42000000** (5 MHz to 42 MHz)
- EuroDOCSIS: **5000000** to **65000000** (5 MHz to 65 MHz)

cable upstream group-id

Syntax: **cable upstream *n* group-id {*g*}**

Specify the upstream group that the upstream belongs to. Valid range: **1** to **6**.

This provides a form of load balancing by distributing cable modems across upstreams with the same group-id during registration according to the cable group policy.

The default group-ids are **1** to **6** for upstreams 1 to 6 respectively, so by default no load balancing occurs.

See also: “cable group…” on page 6-73, “show cable group” on page 6-31.

cable upstream high-power-offset

Syntax: **cable upstream *n* high-power-offset {*offset*}**

Specifies the maximum allowed input power to the CMTS, in dB, above the nominal input power. Cable modems whose input power is higher than this limit are forced to range. The parameter is:

offset

The maximum allowed offset, in 1/10 dB increments. Valid range: **10** to **100**, in steps of 10 (**10**, **20**, and so forth).

See also: “cable upstream low-power-offset” on page 6-140.

cable upstream ingress-cancellation

Syntax: **[no] cable upstream *n* ingress-cancellation**

Turns on upstream ingress cancellation for the specified upstream channel. The **no** form of this command disables ingress cancellation.

> ***Note:*** This is a separately licensed feature and cannot be enabled unless a separate license is purchased.

cable upstream load-interval

Syntax: **cable upstream *n* load-interval {*time*}**

Sets the time, in seconds, to use as an interval for load averaging. on this interface. Valid range: **30** to **600** seconds.

cable upstream low-power-offset

Syntax: **cable upstream *n* low-power-offset {*offset*}**

Specifies the minimum allowed input power to the CMTS, in dB, below the nominal input power. Cable modems whose input power is lower than this limit are forced to range. The parameter is:

offset

The minimum allowed offset, in 1/10 dB increments. Valid range: **–10** to **–100**, in steps of 10 (**10**, **20**, and so forth).

See also: “cable upstream high-power-offset” on page 6-140.

cable upstream minislot-size

Syntax: **cable upstream *n* minislot-size {*m*}**

Specifies the minislot-size in multiples of time-ticks of 6.25 microsecond each tick. Allowed values are **128**, **64**, **32**, **16**, **8**, **4**, **2**, and **1**.

cable upstream modulation-profile

Syntax: **cable upstream *n* modulation-profile {*p*} [channel-type *type*]**

Selects the modulation profile for this upstream. Valid range: **1** to **10**.

The optional **channel-type** parameter sets the modulation scheme; one of: **atdma**, **scdma**, **tdma**, or **tdma&atdma**.

See also: “cable modulation-profile” on page 6-75.

cable upstream periodic-maintenance-interval

Syntax: **cable upstream *n* periodic-maintenance-interval {*p*}**

Sets the periodic ranging interval.

Valid range: **100** to **10000** in 1/100 second intervals.

cable upstream plant-length

Syntax: **cable upstream *n* plant-length {*l*}**

Sets the initial maintenance region size to allow for timing variation across modems separated by this distance.

Valid range: **1** to **160** km.

> ***Note:*** Set the distance to the maximum one-way distance between modems and the C3 in the plant.

cable upstream power-level

Syntax: **cable upstream *n* power-level {*p*} [fixed | auto]**

Sets the target input power level to be used by the CMTS when it ranges modems. It is generally a bad idea to change this parameter.

p

Target power level. The allowable values depend on the channel width:

200 kHz: –16 to +14 dBmV

400 kHz: –13 to +17 dBmV

800 kHz: –10 to +20 dBmV

1600 kHz: –7 to +23 dBmV

3200 kHz: –4 to +26 dBmV

6400 kHz: 0 to +29 dBmV

auto

Re-adjust the configured power level automatically when the symbol rate changes. In auto mode, doubling the symbol rate increases the configured power level by +3dB to maintain constant SNR on the upstream channel. Similarly, halving symbol rate decreases the configured power level by –3dB.

You can reset the configured power level after a symbol rate change, but any subsequent symbol rate change again changes the configured power level.

Note: Any change in the power level results in a change in modem transmit power levels. The power level is still subject to the maximum ranges detailed above.

fixed

Do not perform automatic power level readjustments.

cable upstream pre-equalization

Syntax: **[no] cable upstream *n* pre-equalization**

Enable cable modem pre-equalization. Use the **no** form of this command to disable pre-equalization.

cable upstream range-backoff

Syntax: **cable upstream *n* range-backoff {automatic | *start end*}**

Sets the random backoff window for initial ranging. The parameters are:

automatic

Automatically change the backoff.

start, end

Manually set the backoff. *start* and *end* must be in the range **0** to_**15**; the value for *end* must be higher than *start*.

cable upstream rate-limit

Syntax: **[no] cable upstream *n* rate-limit [use-token-bucket-for-cos]**

Enables rate limiting. Use the **no** form of this command to disable rate limiting. The parameters are:

use-token-bucket-for-cos

Override DOCSIS 1.0 defaults with token bucket rate-limiting.

cable upstream scrambler

Syntax: **[no] cable upstream *n* scrambler**

Enables the upstream scrambler. Use the **no** form of this command to disable the scrambler.

cable upstream short-periodic-maintenance-interval

Syntax: **cable upstream *n* short-periodic-maintenance-interval {*p*}**

Sets the ranging interval used after a parameter change (timing offset, power, etc.). This allows the modem to complete ranging adjustments quickly without waiting for periodic ranging opportunities.

Valid range: **10000** to **40000000** microseconds. Recommended value is **1000000** (1 second).

cable upstream shutdown

Syntax: **[no] cable upstream *n* shutdown**

Disables the upstream. Use the **no** form of this command to enable the upstream.

cable upstream snr-timeconstant

Syntax: **cable upstream *n* snr-timeconstant {*tc*}**

Sets the amount of averaging of the upstream signal-to-noise (SNR) over time. The parameter is:

tc

The amount of averaging desired. Valid range: **0** to **10**.

0—no averaging; the value of the docsIfSigQSignalNoise MIB is the instantaneous value at the time of the request.

10—maximum averaging; provides an average over all time.

cable upstream status

Syntax: **cable upstream *n* status {activate | deactivate}**

Activates or deactivates the upstream channel.

Router Configuration Mode

Use the global command **router rip** to enter router configuration mode.

> ***Note:*** Router configuration requires a license. Contact your ARRIS representative for a license key.

Example:

```
C3(config)#router rip

C3(config-router)#?
auto-summary       - Enable automatic network number summarization
default-information- Control distribution of default information
default-metric     - Set metric of redistributed routes
end                - Exit configuration mode
exit               - Exit Mode / CLI
help               - Display help about help system
multicast          - Enable multicast routing packet support
network            - Enable routing on an IP network
no                 -
passive-interface  - Suppress routing updates on an interface
redistribute       - Redistribute information from another routing protocol
show               - Show system info
timers             - Adjust routing timers
validate-update-source- Perform sanity checks against source address of routing
updates
version            - Set routing protocol version
*scm               - Alias: "show cable modem"

C3(config-router)#
```

auto-summary

Syntax: **[no] auto-summary**

Enables automatic network number summarization. This can reduce the number of networks advertised by the C3.

default-information

Syntax: **[no] default-information originate [always]**

Controls whether the C3 advertises its default route (ie **0.0.0.0/0**) to neighbors. When this is disabled (the default), the C3 learns its default route.

If the **always** keyword is not specified, then this route is advertised only if C3 has a default route.

With **always**, route 0.0.0.0/0 is advertised by the C3 even though the C3 does not have a default route itself. The C3 may have a relevant learned route (i.e. the C3 can still advertise itself as default router to

CPEs which run RIP so they forward traffic to the C3). The C3 could know a more specific route to the destination to deliver traffic and if not, the C3 will drop the traffic.

default-metric

Syntax: **[no] default-metric {m}**

Sets the metric for advertised routes. This is primarily a way to override the default metric for advertised routes. When a connected or static route gets redistributed into an RIP domain, the C3 needs to start to advertise the route to the neighbor in RIP responses. Connected and static routes do not use a metric specification so the C3 needs to know which metric value to associate with them in RIP advertisement. This value is specified by the **default-metric** command.

When a connected or static route gets redistributed into a RIP domain, the C3 needs to start to advertise the route to the neighbor in RIP responses. Connected and static routes do not use a metric specification so the C3 needs to know which metric value to associate with them in RIP advertisement. This value is specified by the **default-metric** command.

Valid range: **1** to **15**. Default: **1**.

multicast

Syntax: **[no] multicast**

Enables or disables multicast of routing updates. When enabled, the C3 multicasts RIP updates to IP address 224.0.0.9; all RIP v2 routers listen for updates on this address. When disabled, the C3 broadcasts updates (required for RIP v1 operation).

network

Syntax: **[no] network {ipaddr} [wildcard] [disable]**

Enables routing on a network. This is the only required router configuration command to start routing.

Use **network 0.0.0.0 255.255.255.255** to enable routing on all interfaces.

Note that *ipaddr* should be a network address of one of the fastethernet interfaces. Use the **no** form of this command to disable routing on a network.

The wildcard is the inverse of a subnet mask; for example if the subnet mask is **255.255.255.0**, use **0.0.0.255** for the wildcard.

Use the **disable** keyword to turn off RIP on a subnet. You can use this to turn off routing for a portion of a subnet noting that this specification may affect more than one sub-interface:

```
network 10.1.0.0 0.0.255.255
!  turn off RIP for this scope
!  noting that more than one interface may match this scope
network 10.1.36.0 0.0.0.255 disable this scope
```

passive-interface

Syntax: **[no] passive-interface {cable 1/0.s | default | fastethernet 0/n.s}**

Suppress routing updates on an interface. The C3 learns routes on this sub-interface but does not advertise routes.

redistribute connected

Syntax: **[no] redistribute connected [metric m]**

Controls whether the C3 advertises subnets belonging to sub-interfaces and are not under configured network scopes.

Example: Use this command to advertise cable sub-interface subnets into an MSO RIP backbone without running RIP on the cable sub-interface itself for security reasons. (do not want to receive or send RIP updates on the cable sub-interface).

redistribute static

Syntax: **[no] redistribute static [metric m]**

Controls whether the C3 advertises static routes.

Redistributed routes use the optionally-specified metric or the default metric if none is specified.

timers basic

Syntax: **timers basic {interval} {invalid} {flush}**

Sets various router-related timers. The parameters are:

interval

The time, in seconds, between basic routing updates (that is, the C3 generates RIP update packets at this interval).

Valid range: **0** to **4294967295** sec. Default: **30** sec.]]

invalid

The time, in seconds, that the C3 continues to use a route without receiving a RIP update packet for that route. After the timer expires, the C3 advertises the route with metric 16 (no longer reachable).

Valid range: **1** to **4294967295** sec.; the time must be at least 3 times longer than the interval timer. Default: **180** seconds.

Flush

The time, in seconds, after which the C3 flushes and stops advertising invalid routes.

Valid range: **1** to **4294967295** sec; the time must be greater than or equal to the invalid timer. Default: **300** seconds.

validate-update-source

Syntax: **[no] validate-update-source**

Enables or disables sanity checks against received RIP updates, based on the source IP address of the packet. This check is disabled by default.

version

Syntax: **version {1 | 2}**

Sets the version of RIP to use over all C3 interfaces.

In most cases, you should use the default (**version 2**). RIP v1 supports only "classful networks," the traditional class A/B/C subnetworks, which have been largely supplanted by classless subnets. RIP v1 summarizes all routes it knows on classful network boundaries, so it is impossible to subnet a network properly via VLSM. Thus, select **version 1** only if the network the C3 is connected to requires it.

7 Managing Cable Modems

This chapter discusses various aspects of cable modem management. Proper management can result in a more efficient and secure network.

Upstream Load Balancing

Load balancing offers the ability to distribute modems in different ways across grouped upstream channels.

Each upstream channel has a “group ID” assigned to it which is used to associate that channel with other upstream channels on the same physical cable. See the **cable group command** family of commands in Chapter 10.

Cable groups thus reflect the physical cable plant layout and specifically the reverse path combining of the plant. All upstream channels in one cable group should be available to a modem that can see any one of these channels.

Each cable group offers two configurations for load balancing:

1. None
2. Initial Numeric

cable group <id> load-balancing none

No load balancing is performed. Modems come online using any upstream in the same group.

cable group <id> load-balancing initial numeric

With this configuration, the number of modems is evenly distributed across the available active channels in the same group. Modems are redirected to the most appropriate upstream during initial ranging. Once a modem comes online it will remain on the same channel until rebooted at which time it may be moved to another channel if appropriate.

What CPE is attached to a modem?

Use the command **show interfaces cable 1/0 modem 0**.

Example:

```
C3#show interfaces cable 1/0 modem 0
SID    Priv bits   Type      State   IP address       method
1      0           modem     up      10.30.75.143     dhcp
1      0           cpe       unknown 10.30.75.207     dhcp
```

Using ATDMA Upstreams

Several steps must be undertaken to use a DOCSIS 2.0 modem in ATDMA mode on a C3 upstream.

- Configure an ATDMA capable modulation profile in the C3.
- Configure the upstream with a modulation profile containing ATDMA burst descriptors.
- Configure the Upstream channel type for ATDMA operation.

Setting the Configuration File

Give the modem a DOCSIS 1.1 configuration file with the following TLV added to it for a DOCSIS 2.0 modem to use an ATDMA capable upstream.

Paramteter	Value
Type	39
Length	1
Value	1 for DOCSIS 2.0

Note: The above parameters are the defaults. A DOCSIS 2.0 cable modem should assume this setting if not specified.

Configuring a Modulation Profile

The C3 has a short-cut method for creating an ATDMA modulation profile. Create a new modulation profile using the commands:

```
conf t
cable modulation-profile 3 advanced-phy
```

Assign the new modulation profile to the required upstream using the command sequence:

```
int ca 1/0
cable upstream 0 modulation-profile 3
exit
```

The following is an example modulation profile created using the above commands:

```
cable modulation-profile 3 request AdvPhy preamble-type qpsk0
cable modulation-profile 3 request 0 16 0 8 qpsk scrambler 338 no-diff 64 fixed
cable modulation-profile 3 initial AdvPhy ATDMA 1 1536
cable modulation-profile 3 initial AdvPhy preamble-type qpsk0
cable modulation-profile 3 initial 5 34 0 48 qpsk scrambler 338 no-diff 640
fixed
cable modulation-profile 3 station AdvPhy ATDMA 1 1536
cable modulation-profile 3 station AdvPhy preamble-type qpsk0
cable modulation-profile 3 station 5 34 0 48 qpsk scrambler 338 no-diff 384
fixed
cable modulation-profile 3 advPhyS AdvPhy ATDMA 1 1536
cable modulation-profile 3 advPhyS AdvPhy preamble-type qpsk1
cable modulation-profile 3 advPhyS 12 78 14 8 64qam scrambler 338 no-diff 104
fixed
cable modulation-profile 3 advPhyL AdvPhy ATDMA 1 1536
cable modulation-profile 3 advPhyL AdvPhy preamble-type qpsk1
cable modulation-profile 3 advPhyL 16 220 0 8 64qam scrambler 338 no-diff 104
fixed
cable modulation-profile 3 advPhyU AdvPhy ATDMA 1 1536
cable modulation-profile 3 advPhyU AdvPhy preamble-type qpsk1
cable modulation-profile 3 advPhyU 16 220 0 8 64qam scrambler 338 no-diff 104
fixed
```

Changing the Upstream Channel Type

Use the command cable upstream 0 channel-type atdma to change the upstream channel type.

DHCP

Dynamic Host Configuration Protocol (DHCP) is used by cable modems, and CPE devices attached to the cable modem, to obtain both an IP address and initial operating parameters. This parameter or "option" transfer is the first interaction a cable modem has with management systems beyond the CMTS.

DHCP traffic between the DHCP server and the clients (cable modems and subscriber CPEs) travel through the C3. The C3 in turn can either pass the traffic through or take a more active role.

You have two options:

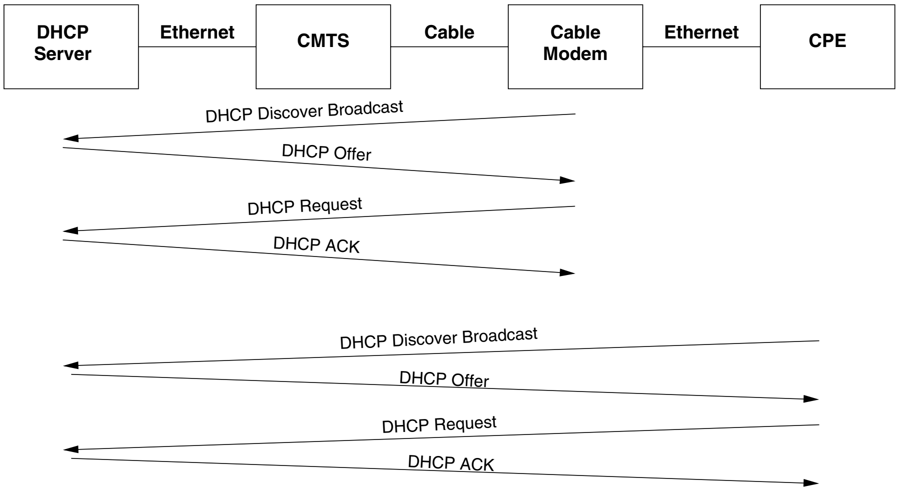
- Transparent mode (the default): the C3 re-broadcasts DHCP broadcast packets received from a cable sub-interface to all active fastethernet sub-interfaces in the same bridge group. Transparent mode requires that the DHCP server must be within the same subnet as the CPE.
- DHCP relay mode: by specifying **ip dhcp relay** on a cable sub-interface, the C3 can reduce broadcast traffic by sending DHCP broadcast packets only to specific fastethernet sub-interfaces.

  Note: DHCP relay is required for routing sub-interfaces.

The following sections describe each mode.

Transparent Mode

The first option, transparent mode, is the factory default. In this case the C3 simply passes DHCP messages along and takes no part in the DHCP process. The following diagram shows the flow of DHCP traffic through the C3 in transparent mode.



DHCP Relay Mode

When DHCP Relay is active on a cable sub-interface, the C3 intercepts DHCP broadcast packets received at the cable sub-interface and re-directs them to all fastethernet sub-interfaces, or to a specific address if you specify **cable helper-address**.

You activate DHCP Relay on specific cable sub-interfaces using the **ip dhcp relay** command in interface configuration mode; there are also several options that can be activated individually on each sub-interface. The sections following describe these options and their uses.

What Happens During Relay

The C3 knows the difference between a cable modem and a CPE device and can:

- direct DHCP as a unicast to specific DHCP servers based on whether the DHCP message is coming from a cable modem or an attached host using the cable interface configuration command:

  cable helper-address {*ipaddr*} [cable-modem | host]

- assist the DHCP server to allocate different IP address spaces to cable modems and CPE devices using the cable interface configuration command:

  cable dhcp-giaddr policy

- assist the subscriber management systems by telling the DHCP server what cable modem a host (CPE) is attached to and identifying a CPE device attached to a cable modem by using the cable interface configuration command:

ip dhcp relay information option

- DHCP unicast (renew) is intercepted and forwarded—not bridged—to the required destination address regardless of the CPE or CM default route settings.

  Where the destination address (or the gateway to the destination address) is not directly connected to a bridge group the unicast renew was received in, the unicast will be forwarded across bridge groups to the required interface but **l2-bg-to-bg-routing** must be activated in all the involved bridge groups for any ack to a DHCP RENEW to be forwarded back to the originating bridge-group.

Directing DHCP Broadcasts to Specific Servers

The most useful functions of the **cable helper-address** command are:

- To change the broadcast DHCP message arriving at the cable sub-interface to a unicast message leaving the C3 directed to a specific DHCP server.
- To allow the DHCP server to exist on a routed backbone. The DHCP discover messages from cable-modems or hosts are now uni-cast to the specified DHCP server. Where routers are between the DHCP servers and the C3 (the DHCP server IP subnet is not known to the C3), the use of static routes using the "ip route" command in the C3 may be required or "router rip" activated.
- In bridging mode, DHCP can be forwarded across bridge groups.

  Where the helper address (or the gateway to the helper address) is not directly connected to a bridge group the broadcast was received in, the C3 forwards the unicast across bridge groups to the required interface, but **l2-bg-to-bg-routing** must be activated in all the involved bridge groups for any reply to this message to be forwarded back to the originating bridge group.

If no helper address is specified, the C3 bridges the broadcast to all FastEthernet sub-interfaces in the same bridge group, or drops the packet if no bridge group membership exists (such as on a routed sub-interface).

If the helper address is not within a subnet known to the C3, the C3 inspects its IP route table for a route to this destination subnet—this route then specifies the sub-interface to use for the unicast. If such a route does not exist, no unicast will occur.

The routing table can be influenced by:

- primary and secondary IP addresses of sub-interfaces and the resulting subnet memberships of those interfaces
- **ip default-gateway** specification in bridging mode
- **ip route 0.0.0.0 0.0.0.0 a.b.c.d** specification for the route of last resort in IP routing mode
- a static route configured with **ip route**
- RIP propagation in the network

The C3 can differentiate between DHCP messages from cable modems and hosts. The **cable helper-address** command allows such DHCP messages to be directed to different DHCP servers.

Example:

The cable operator manages the cable-modem IP addresses , an ISP manages the host IP addresses.

```
cable 1/0.0
cable helper-address 10.1.1.1 cable-modem
cable helper-address 10.2.2.2 host
```

Up to 5 helper-addresses may be specified per helper address classification (modem, host, or either). Only the DHCP helper-addresses of the sub-interface the DHCP message is received on are used.

Example 1:

```
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.0

interface Cable 1/0.0
 cable helper-address A cable-modem
 cable helper-address B cable-modem
 cable helper-address C
 cable helper-address D
 cable helper-address E
```

The C3 sends any cable modem’s DHCP discover/request to helper addresses A and B, and any host’s DHCP discover/request to helper addresses C, D and E.

Example 2:

```
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.0

interface Cable 1/0.0
 cable helper-address A host
 cable helper-address B host
 cable helper-address C host
 cable helper-address D
 cable helper-address E
```

Any cable modem’s DHCP discover/request will be sent to helper addresses D and E. Any host’s DHCP discover/request will be sent to helper addresses A, B and C.

Example 3:

```
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.0

interface Cable 1/0.0
 cable helper-address A cable-modem
 cable helper-address B host
 cable helper-address C host
 cable helper-address D
 cable helper-address E
```

Any cable modem’s DHCP discover/request is sent to helper address A. Any host’s DHCP discover/request will be sent to helper addresses B and C. Helper addresses D and E are redundant in this configuration.

See “cable helper-address” on page 6-133 for syntax and other information.

Redundant DHCP server support

Where multiple helper-addresses are specified, the C3 unicasts the DHCP Discover to each of the specified helper addresses. Any ensuing communication with the DHCP client is unicast only to the DHCP server that responded to the first DHCP Discover unicast. If a subsequent DHCP request is not answered by this DHCP server, the C3 again unicasts the message to all specified DHCP servers.

cable helper-address a.b.c.d

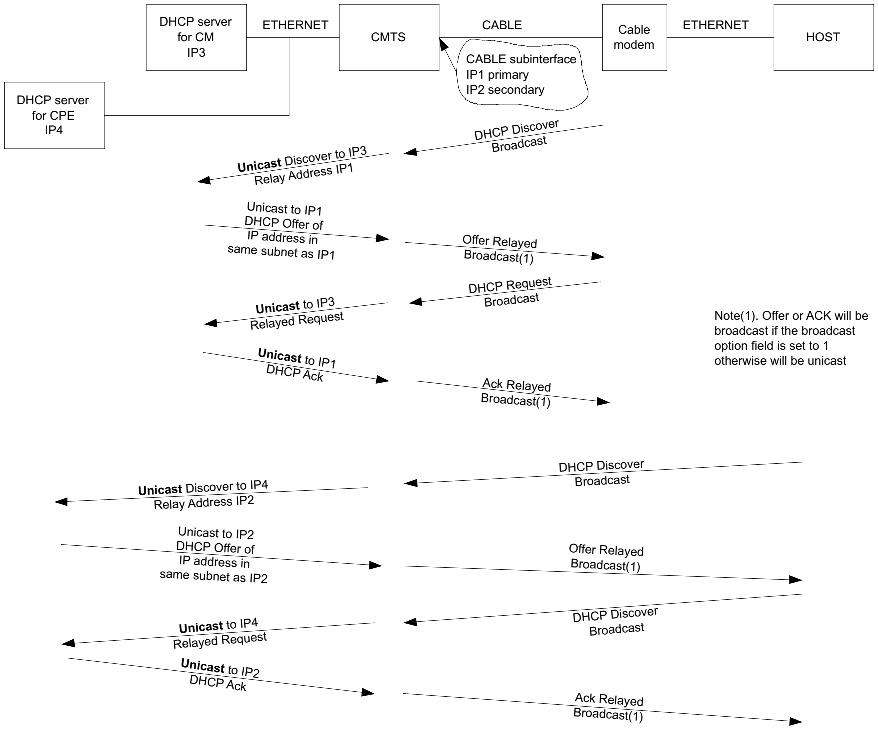
unicasts all DHCP broadcast messages to the specified DHCP server IP address

cable helper-address a.b.c.d cable modem

unicasts all cable modem generated DHCP broadcast messages to the specified DHCP server IP address

cable helper-address a.b.c.d host

unicasts all host generated DHCP broadcast messages to the specified DHCP server IP address


DHCP server for CM IP3
ETHERNET
CMTS
CABLE
Cable modem
ETHERNET
HOST
CABLE subinterface IP1 primary IP2 secondary
DHCP server for CPE IP4
DHCP Discover Broadcast
Unicast Discover to IP3 Relay Address IP1
Unicast to IP1 DHCP Offer of IP address in same subnet as IP1
Offer Relayed Broadcast(1)
DHCP Request Broadcast
Unicast to IP3 Relayed Request
Note(1). Offer or ACK will be broadcast if the broadcast option field is set to 1 otherwise will be unicast
Unicast to IP1 DHCP Ack
Ack Relayed Broadcast(1)
DHCP Discover Broadcast
Unicast Discover to IP4 Relay Address IP2
Unicast to IP2 DHCP Offer of IP address in same subnet as IP2
Offer Relayed Broadcast(1)
DHCP Discover Broadcast
Unicast to IP4 Relayed Request
Unicast to IP2 DHCP Ack
Ack Relayed Broadcast(1)

Verifying DHCP Forwarding

DHCP forwarding operation can be verified using the C3 debug facilities.

> ***Note:*** If debugging CPE DHCP, turn on debug for the MAC address of the modem that the CPE is attached to.

For example, use the following commands from privilege mode.

```
terminal monitor
debug cable dhcp-relay

debug cable mac-address 00A0.7374.BE70

16:51:34: DHCPRELAY: DISCOVER: adding relay information option
16:51:34: DHCPRELAY: DISCOVER: setting giaddr to 10.250.139.2
16:51:34: DHCPRELAY: DISCOVER: from 00A0.7374.BE70 forwarded to
10.250.139.1
16:51:34: DHCPRELAY: OFFER: Removing information option from
frame
```

```
16:51:34: DHCPRELAY: Broadcasting OFFER to client
00A0.7374.BE70
16:51:37: DHCPRELAY: REQUEST: adding relay information option
16:51:37: DHCPRELAY: REQUEST: setting giaddr to 10.250.139.2
16:51:37: DHCPRELAY: REQUEST: from 00A0.7374.BE70 forwarded to
server 10.250.139.1
16:51:37: DHCPRELAY: ACK: Removing information option from frame
16:51:37: DHCPRELAY: Broadcasting ACK to client 00A0.7374.BE70
```

```
debug cable mac-address 00A0.7374.BE70 verbose
```

```
16:54:29: DHCPRELAY: DISCOVER: adding relay information option
16:54:29: DHCPRELAY: DISCOVER: from 00A0.7374.BE70 forwarded to
10.250.139.1
16:54:29: DHCPRELAY: Dumping outgoing UDP packet:
 01 01 06 01 73 74 BE 70 00 00 80 00 00 00 00 00
 00 00 00 00 00 00 00 00 0A FA 8B 02 00 A0 73 74
 BE 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 01 3C 56 64 6F 63 73 69 73 31 2E 31 3A 30
 35 32 34 30 31 30 31 30 31 30 32 30 31 30 31 30
 33 30 31 30 31 30 34 30 31 30 31 30 35 30 31 30
 31 30 36 30 31 30 31 30 37 30 31 31 30 30 38 30
 31 31 30 30 39 30 31 30 30 30 61 30 31 30 31 30
 62 30 31 30 38 30 63 30 31 30 31 3D 07 01 00 A0
 73 74 BE 70 39 02 02 40 37 07 01 1C 43 03 02 04
 07 52 14 01 04 80 00 00 03 02 06 00 A0 73 74 BE
 70 04 04 00 00 00 00 FF 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00


16:54:29: DHCPRELAY: Dumping incoming UDP packet:
 02 01 06 00 73 74 BE 70 00 00 80 00 00 00 00 00
 0A FA 8B 0E 0A FA 8B 01 0A FA 8B 02 00 A0 73 74
 BE 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 76 6C 61 6E
 5F 34 32 2E 63 66 67 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 02 36 04 0A FA 8B 01 33 04 00 07 A9 33 01
 04 FF FF FF 00 06 08 C0 A8 FA C2 C0 A8 FA C3 2C
 04 C0 A8 FA C2 1C 04 FF FF FF FF 03 04 0A FA 8B
 01 52 14 01 04 80 00 00 03 02 06 00 A0 73 74 BE
 70 04 04 00 00 00 00 FF
16:54:29: DHCPRELAY: OFFER: Removing information option from
frame
16:54:29: DHCPRELAY: Broadcasting OFFER to client
00A0.7374.BE70
16:54:29: DHCPRELAY: Dumping outgoing UDP packet:
 02 01 06 00 73 74 BE 70 00 00 80 00 00 00 00 00
 0A FA 8B 0E 0A FA 8B 01 0A FA 8B 02 00 A0 73 74
 BE 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 76 6C 61 6E
 5F 34 32 2E 63 66 67 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 02 36 04 0A FA 8B 01 33 04 00 07 A9 33 01
 04 FF FF FF 00 06 08 C0 A8 FA C2 C0 A8 FA C3 2C
 04 C0 A8 FA C2 1C 04 FF FF FF FF 03 04 0A FA 8B
 01 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00
```

```
16:54:30: DHCPRELAY: Dumping incoming UDP packet:
 01 01 06 00 73 74 BE 56 00 00 80 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 A0 73 74
 BE 56 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 03 3C 56 64 6F 63 73 69 73 31 2E 31 3A 30
 35 32 34 30 31 30 31 30 31 30 32 30 31 30 31 30
 33 30 31 30 31 30 34 30 31 30 31 30 35 30 31 30
 31 30 36 30 31 30 31 30 37 30 31 31 30 30 38 30
 31 31 30 30 39 30 31 30 30 30 61 30 31 30 31 30
 62 30 31 30 38 30 63 30 31 30 31 3D 07 01 00 A0
 73 74 BE 56 32 04 0A FA 8B 6C 36 04 0A FA 8B 01
 39 02 02 40 37 07 01 1C 43 03 02 04 07 FF 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00
16:54:31: DHCPRELAY: REQUEST: adding relay information option
16:54:31: DHCPRELAY: REQUEST: from 00A0.7374.BE70 forwarded to
server 10.250.139.1
16:54:31: DHCPRELAY: Dumping outgoing UDP packet:
 01 01 06 01 73 74 BE 70 00 00 80 00 00 00 00 00
 00 00 00 00 00 00 00 00 0A FA 8B 02 00 A0 73 74
 BE 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 03 3C 56 64 6F 63 73 69 73 31 2E 31 3A 30
 35 32 34 30 31 30 31 30 31 30 32 30 31 30 31 30
 33 30 31 30 31 30 34 30 31 30 31 30 35 30 31 30
 31 30 36 30 31 30 31 30 37 30 31 31 30 30 38 30
 31 31 30 30 39 30 31 30 30 30 61 30 31 30 31 30
 62 30 31 30 38 30 63 30 31 30 31 3D 07 01 00 A0
 73 74 BE 70 32 04 0A FA 8B 0E 36 04 0A FA 8B 01
 39 02 02 40 37 07 01 1C 43 03 02 04 07 52 0E 01
 04 80 00 00 03 02 06 00 A0 73 74 BE 70 FF 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00


16:54:31: DHCPRELAY: Dumping incoming UDP packet:
 02 01 06 00 73 74 BE 70 00 00 80 00 00 00 00 00
 0A FA 8B 0E 0A FA 8B 01 0A FA 8B 02 00 A0 73 74
 BE 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 76 6C 61 6E
 5F 34 32 2E 63 66 67 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 05 36 04 0A FA 8B 01 33 04 00 07 A9 30 01
 04 FF FF FF 00 06 08 C0 A8 FA C2 C0 A8 FA C3 2C
 04 C0 A8 FA C2 1C 04 FF FF FF FF 03 04 0A FA 8B
 01 52 0E 01 04 80 00 00 03 02 06 00 A0 73 74 BE
 70 FF
```

```
16:54:31: DHCPRELAY: ACK: Removing information option from frame
16:54:31: DHCPRELAY: Broadcasting ACK to client 00A0.7374.BE70
16:54:31: DHCPRELAY: Dumping outgoing UDP packet:
 02 01 06 00 73 74 BE 70 00 00 80 00 00 00 00 00
 0A FA 8B 0E 0A FA 8B 01 0A FA 8B 02 00 A0 73 74
 BE 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 76 6C 61 6E
 5F 34 32 2E 63 66 67 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
 35 01 05 36 04 0A FA 8B 01 33 04 00 07 A9 30 01
 04 FF FF FF 00 06 08 C0 A8 FA C2 C0 A8 FA C3 2C
 04 C0 A8 FA C2 1C 04 FF FF FF FF 03 04 0A FA 8B
 01 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00
```

Relay Agent Support

The C3 can modify the DHCP relay address information (giaddr field) in the DHCP messages from the cable modem or host.

The primary function of this DHCP field is to allow the DHCP Offer and DHCP Ack to be routed back to the requesting device through what may be many routers in the backbone network. The giaddr advertises the C3 as the gateway to the requesting device.

DHCP servers use this relay address as a hint to what address space programmed into the DHCP server (address scope) to allocate an address from.

The DHCP server looks at the relay address and searches its defined scopes looking for a subnet match. If a matching scope is found, it allocates a lease from that scope.

The following example uses the interface’s secondary address to specify the host giaddr:

```
cable 1/0.0
ip address 10.1.1.1 255.255.255.0
ip address 10.2.2.1 255.255.255.0 secondary
ip dhcp relay
! use same DHCP server for host and cable-modems
cable helper-address 10.9.9.1
```

```
! update giaddr with 10.1.1.1 for modems
! update giaddr with 10.2.2.1 for hosts
cable dhcp-giaddr policy
```

If **cable dhcp-giaddr policy** is activated, the cable sub-interface used on the C3 to relay the DHCP (as dictated by **cable helper-address** and **ip route**) should be configured with a *secondary* IP address. Otherwise the C3 uses the primary IP address as the giaddr (even with **dhcp-giaddr policy** activated).

The following example uses VSE encoding and cable sub-interfaces to specify the host giaddr:

```
cable 1/0.0
! one subnet used for all cable modem access
ip address 10.1.1.1 255.255.255.0
ip dhcp relay
cable helper-address 10.9.9.1 cable-modem
cable dhcp-giaddr primary

cable 1/0.2
! VSE modems with tag 2 will have attached CPE
! mapped to this sub-interface
ip address 10.2.2.1 255.255.255.0
encapsulation dot1q 2 native
! use the primary sub-interface address for host giaddr
ip dhcp relay
cable helper-address 10.9.9.1 host
cable dhcp-giaddr primary

cable 1/0.3
! VSE modems with tag 3 will have attached CPE
! mapped to this sub-interface
ip address 10.3.3.1 255.255.255.0
encapsulation dot1q 3 native
! use the primary sub-interface address for host giaddr
ip dhcp relay
cable helper-address 10.9.9.1 host
cable dhcp-giaddr primary
```

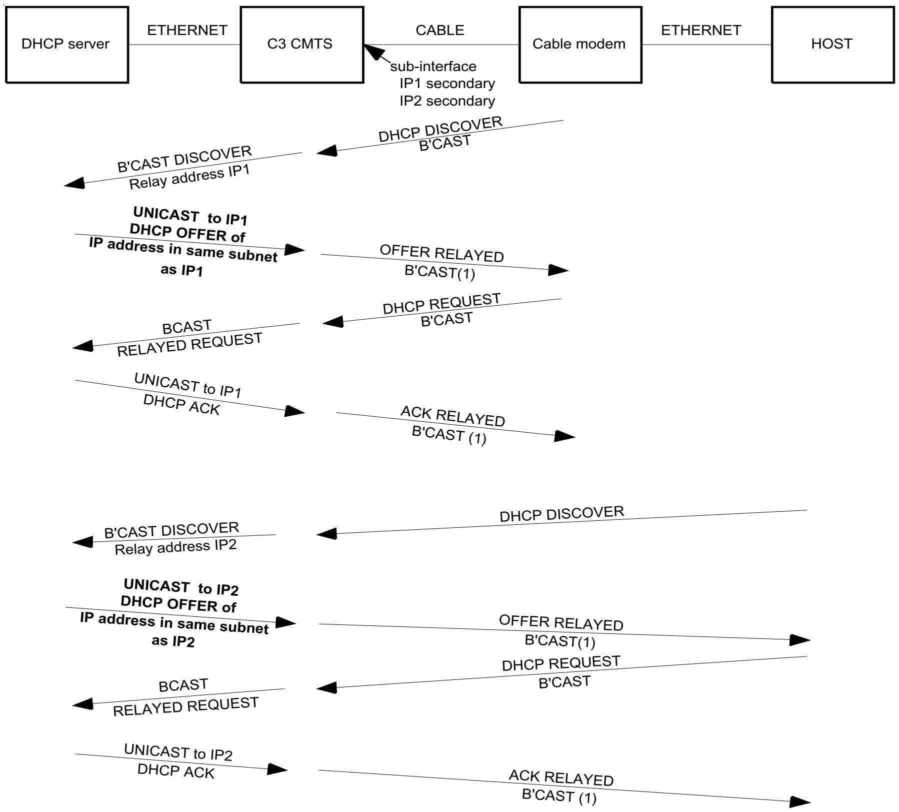
The following examples uses **map-cpes** and cable sub-interfaces to specify the host giaddr:

```
cable 1/0.0
! subnet used for cable modem DHCP access only
ip address 10.1.1.1 255.255.255.0
ip dhcp relay
cable helper-address 10.9.9.1 cable-modem
cable dhcp-giaddr primary
cable 1/0.2
! modems given 10.2.2.0 address will come here
ip address 10.2.2.1 255.255.255.0
encapsulation dot1q 2 native
map-cpes cable 1/0.12

cable 1/0.3
! modems given 10.3.3.0 address will come here
ip address 10.3.3.1 255.255.255.0
encapsulation dot1q 3 native
map-cpes cable 1/0.13

cable 1/0.12
! CPE mapped to this sub-interface
ip address 10.12.12.1 255.255.255.0
encapsulation dot1q 12 native
ip dhcp relay
cable helper-address 10.9.9.1 host
! use the primary sub-interface address for host giaddr
cable dhcp-giaddr primary

cable 1/0.13
! CPE mapped to this sub-interface
ip address 10.13.13.1 255.255.255.0
encapsulation dot1q 13 native
ip dhcp relay
cable helper-address 10.9.9.1 host
! use the primary sub-interface address for host giaddr
cable dhcp-giaddr primary
```

If **cable helper-address** is not being used:

- If the sub-interface is Layer 3, then the DHCP message will be dropped; a cable helper-address is mandatory for Layer 3 Cable sub-interfaces that have DHCP Relay activated.
- If the sub-interface is Layer 2, then C3 broadcasts the DHCP message with updated giaddr from every active fastethernet sub-interface in the same bridge group.

The following diagram shows DHCP traffic flow with **dhcp-giaddr** enabled:



DHCP Relay Information Option

The C3 can insert an option (option number 82) in the DHCP Discover or Request message that tells the management systems at the time of cable modem (or host) DHCP whether the DHCP is from a modem or a host. The MAC address of the cable modem is inserted into this option field for every DHCP Discover or Request message (with the exception of Renews) relayed by the C3 from the cable plant.

If the MAC address in the chaddr field matches the MAC address stored in the option 82 field, the discover or request must have come from a cable modem.

Similarly, if the MAC addresses do not match, then the Discover or Request can be assumed to have:

- come from a host, and
- the host is attached to the cable modem identified by the MAC address in the option 82 agent-remote-id suboption (sub-option 2) field.

DHCP Server Use of Option 82

A DHCP server searches its defined scopes for a match to the giaddr of the incoming DHCP Discover or Request. (If the DHCP Discover or Request arrives as a broadcast, then the giaddr is assumed to be that of the received sub-interface IP address). If a matching scope is found, a reserved address is looked for in this scope. If no reserved address is found, then the next available IP address in this scope will be leased: that is, the leased address is always within the same subnet as the giaddr.

Where one modem subnet is required, this is not a problem. Where modems are required to be in different subnets, this is a problem. The DHCP server must be forced to lease an address in a different scope to the scope that matches the giaddr.

DHCP servers allow this to occur in different ways:

- For example Windows 2000 server DHCP server allows a *super scope* to be defined containing a number of scopes. In this case the super scope is searched for a matching scope to the giaddr; if a matching scope is found, the super scope is deemed to be a match. Then a reserved address is looked for. The reserved address can be in any scope in the super scope and does not have to be in the same subnet as the incoming giaddr. If no reserved address is found, then an address is leased on a round robin basis from any of the scopes in the super scope.
- Cisco Network Registrar operates in a similar manner. CNR uses the concept of *primary* and *secondary* scopes. One primary scope may have many secondary scopes. Together the primary and secondary scopes form a super scope in the Windows DHCP server sense.

To summarize DHCP server behavior:

- Where one scope only exists for a giaddr, either a reserved address is issued or an available address from this scope is issued.
- Where two scopes exist and an address is reserved in one scope, but the incoming giaddr matches the DHCP discover to the other scope, the reserved address is not issued. Further, no address from the scope matching the giaddr is issued.
- If the two scopes are a member of a super scope or are in a primary/secondary relationship, the reserved address is issued and if no reserved address is present, an address from either scope is issued on a round robin basis.

The main aim of DOCSIS provisioning is to reserve the MAC address of a modem in a scope, but not to have to do this for a PC. Option 82-aware DHCP servers can assist in this process.

Introducing a concept of primary and secondary DHCP clients:

- A *primary client* has a DHCP Discover with the chaddr field matching the option 82 agent-remote-id suboption field (sub-option number 2).
- A *secondary client* has different MAC addresses in each of these fields and the option 82 agent-remote-id sub-option field (sub-option number 2) is the MAC address of the attached primary device.

When a DHCP Discover arrives from a primary device, all primary scopes are searched as per normal DHCP server operation and either a reserved address issued from a scope matching the giaddr or the next available address is issued from the primary scope matching the giaddr.

When a DHCP Discover arrives from a secondary device, the primary leases are searched for the attached primary MAC address. The lease then defines the primary scope used to issue the primary device IP address. Then the scopes secondary to this primary scope are searched for a reserved address. If no reserved address is found, the next available lease from the secondary scope is issued.

Note: A giaddr match is not performed to the secondary scope.

It is possible to have many secondary scopes to the one primary scope. If no reserved lease is found, then the next available lease from any one of the secondary scopes can be issued on a round robin basis.

Thus once the primary device is allocated an IP address, the secondary device is automatically allocated an IP address from a secondary scope with no need to reserve the address of the secondary device or no need to have a matching giaddr scope for the secondary device.

A side benefit of option 82 processing in a DHCP server is that if no option 82 information is present in the DHCP Discover or Request, primary and secondary scope processing still occurs but slightly differently.

Now the giaddr is used to search all defined scopes. If a matching scope is found but this scope has secondary scopes defined, the secondary scopes are searched for an address reservation. If no reservation is found, an address is issued from the primary and secondary scopes on around robin basis. This operation is very similar to the Windows 2000 server concept of super scopes.

With particular reference to the C3:

When operating in VSE mode, all modems exist in the one subnet and thus are assigned an address from the one scope.

The main requirement on the DHCP server is that modems are able to be given individual DHCP options that override the options normally associated with the scope. In this case, the different option of concern is the configuration file to be given to the modem.

Assuming the DHCP server supports this feature, CPEs are mapped to sub-interfaces by the modem configuration file VSE encoding.

CPEs subsequently perform DHCP using a giaddr of the mapped cable sub-interface. Where a single CPE scope is to be used, the CPE is issued an IP address based on the giaddr—an IP address of this cable sub-interface.

Where multiple CPE subnets are to be used (as in the case of an ISP having multiple non-contiguous or small subnets), the Windows DHCP server "super scope" or CNR's "primary + secondary" processing can be used to issue an IP address from the available scopes on a round robin basis.

- Windows 2000: The giaddr scope is just one scope of many in a super scope—an address is issued on a round robin basis from any of the scopes in the matching super scope.
- Cisco CNR: The giaddr scope matches at least one scope in a primary/secondary set of scopes —an address is issued from the primary and secondary scopes on a round robin basis.

Managing Modems Using SNMP

Simple Network Management Protocol (SNMP) enables you to monitor and control network devices in DOCSIS systems, and to manage configurations, statistics collection, performance, and security. SNMPv2c is used throughout DOCSIS. It supports centralized as well as distributed network management strategies, and includes improvements in the Structure of Management Information (SMI), protocol operations, management architecture, and security. The C3 also supports SNMPv3 for greater network security.

The configuration options available are defined in the **snmp-server** series of global configuration commands, starting on page 6-100.

By using an SNMP Manager application, such as HP OpenView, SNMPc, or NET-SNMP, you can monitor and control devices on the cable network using MIB variables.

> ***Note:*** SNMP access to the CMTS is off by default. You can set up basic access using the following global configuration commands:
>
> ```
> snmp-server community public ro
> snmp-server community private rw
> ```

MIB Variables

Management information is a collection of managed objects, or variables, that reside in a virtual information store called the Management Information Base (MIB). Collections of related objects are defined in MIB modules.

MIB objects are defined by a textual name and a corresponding object identifier, syntax, access mode, status, and description of the semantics of the managed object.

The following shows the format of a DOCSIS MIB variable.

```
docsIfDownChannelPower OBJECT-TYPE
           SYNTAX      TenthdBmV
           UNITS       "dBmV"
           MAX-ACCESS  read-write
           STATUS      current
           DESCRIPTION
               "At the CMTS, the operational transmit power. At the CM,
                the received power level. May be set to zero at the CM
                if power level measurement is not supported.
                If the interface is down, this object either returns
                the configured value (CMTS), the most current value (CM)
                or the value of 0. See the associated conformance object
                for write conditions and limitations. See the reference
                for recommended and required power levels."
           REFERENCE
               "DOCSIS Radio Frequency Interface Specification,
                Table 4-12 and Table 4-13."
```

For a complete list of the current DOCSIS MIBs, see the **Cablelabs website** at (http://www.cablelabs.com/).

Configuring a Host as a Trap Listener

The following CLI commands register the host 192.168.250.107 as a SNMPv2c trap listener. Traps sent to this listener have 'MyCommunity' as a community string and only traps registered under the 'internet' domain are sent (which are basically all traps that a CMTS would send).

Each command requires a unique identifier for each trap listener. You should replace the 'My' prefix with a proper unique identifier, such as a host name.

```
C3# configure terminal
C3(config)# snmp-server user MyCommunity MyGroup v2c access-list Trap
C3(config)# snmp-server group MyGroup v2c notify MyTrapNotify
C3(config)# snmp-server view MyTrapNotify internet included
C3(config)# snmp-server notif-sec-model MySecurity MyCommunity v2c security-
model v2
C3(config)# snmp-server host MyTrapReceiver MySecurity 192.168.250.107 traps
C3(config)# snmp-server enable traps
```

> ***Note:*** Use the command **show snmp-server** to list these settings. These settings are persistent across reboots.

Controlling User Access

You can control access to the network using password-like community strings that enable you to assign users to communities that have names (for example, public or private). This system enables you to manage devices on the network. Community names should be kept confidential.

To prevent unauthorized users from accessing the modem, you assign the modem to a community. You can also specify that SNMP access is allowed only from the cable side. You assign a modem to a community using the docsDevNmAccess group MIBs from either a MIB Browser in an SNMP manager, or by specifying the MIB in the configuration file.

Checking Modem Status

The following table lists useful MIBs for checking the status of a modem using SNMPv2.

General Modem Status

Use the following MIB to check general modem status.

MIB Object	Value	Description
docsIfCmStatusValue	2=notReady	Modem is searching for a downstream channel.
	3=notSynchronized	Modem has found a downstream channel but has not set timing.
	4=phySynchronized	Modem sees a digital signal and is looking for a UCD.
	5=usParameters-Acquired	Modem has found a UCD and is ranging.
	6=rangingComplete	Modem is waiting for a DHCP address.
	7=ipComplete	Modem has IP address and is trying to contact a Network Time Protocol (NTP) server.
	8=todEstablished	Modem has determined the time.
	9=securityEstablished	
	10=paramTransfer-Complete	Received the configuration file.
	11=registration-Complete	CMTS accepted the registration request.
	12=operational	Modem is online.
	13=accessDenied	CMTS does not allow modem to pass traffic.

Data Errors

Use the following MIBs to check for data errors.

MIB Object	Description
docsIfSigQUnerroreds	Number of data packets that arrived undamaged.
docsIfSigQCorrecteds	Number of data packets that arrived damaged, but could be corrected.

MIB Object	Description
docsIfSigQUncorrectables	Number of data packets that arrived so damaged that they were discarded.

Signal-to-Noise Ratio

Use the following MIB to determine the downstream signal-to-noise ratio as measured at the cable modem.

MIB Object	Value	Description
docsIfSigQSignalNoise	35 to 37	Typical ratio for clean plant.
	Below 29	QAM256 is not usable.
	Below 26	QAM64 performance is significantly impaired.
	20	Modem cannot function.

Downstream Channel

Use the following MIBs to determine downstream channel issues.

MIB Object	Value	Description
docsIfCmStatus-LostSyncs	should be small	Number of times modem detects downstream had trouble. A high number indicates problems on the downstream.
docsIfDownChannel-Frequency		Downstream frequency to which the modem is listening.
docsIfDownChannel-Width	6MHz or 8MHz	Set automatically based on whether the CMTS is operating in DOCSIS or EuroDOCSIS mode.
DocsIfDownChannel-Modulation	QAM64 or QAM 256	If different, modem has problem.
DocsIfDownChannel-Power	> +15 dBmv	Signal is too strong; insert an attenuator.
	< -15 dBmv	Signal is too weak; modem might have reliability problems, such a bad cable, too many splitters, or unnecessary attenuator.
	+15 dBmv to -15 dBmv	Valid DOCSIS range.

Upstream Channel

Use the following MIBs to determine upstream channel issues.

MIB Object	Value	Description
docsIfUpChannel-Frequency	should be small	This variable is set automatically by the modem when it selects a particular upstream to use.
docsIfUpChannelWidth		The wider the upstream channel is, the higher the data rate.
docsIfCmStatusTx-Power	+8 to +58 dBmv	Legal range.
	Over +50 dBmv	Do not use 16 QAM; upstream is impaired to the point where QPSK is required.

Upgrading Modem Firmware

Inspecting and upgrading modem firmware is a fundamental part of managing modem operations.

Action

Perform any of the following procedures as necessary.

Upgrading from the Configuration File

1 Using a configuration editor, modify the following fields in the cable modem configuration file:

a In the Software Upgrade Filename field, enter the path and filename of the firmware that you want to download.

b In the SNMP MIB Object field, enter the following hex string: **30 0F 06 0A 2B 06 01 02 01 45 01 03 03 00 02 01**

This hex string sets the docsDevSwAdminStatus variable (MIB object ID **1.3.6.1.2.1.69.1.3.3.0**) to the integer value **2** which allows the modems to perform the upgrade.

c In the Software Upgrade TFTP Server, type the IP address of the TFTP server where the upgrade file is located.

2 Save your changes to the configuration file.

3 Reboot the modems.

Upgrade a Single Modem Using an SNMP Manager

1 Type the IP address of the cable modem in the Name or IP Address field.

2 Type **private** (or the proper Set Community name) in the Community field.

3 Highlight the docsDevMIBObjects MIB (MIB Object ID **1.3.6.1.2.1.69.1**), then click **Down Tree**.

4 Highlight the docsDevSoftware MIB, then click **Down Tree**.

5 From the MIB Values field, highlight **docsDevSwServer**.

6 From the SNMP Set Value field, type the IP address of the TFTP server, then click **Set**.

7 Click **Close** on the pop-up information screen.

8 From the MIB Values field, highlight **docsDevSwFilename**.

9 From the SNMP Set Value field, type the location and filename of the image, then click **Set**.

10 Click **Close** on the pop-up information screen.

11 From the MIB Values field, highlight **docsDevSwAdminStatus**.

12 From the SNMP Set Value field, type **1** (**upgradeFromMgt**), then click **Set**.

13 From the MIB Values field, highlight **docsDevSwOperStatus**.

14 Click **Start Query** to verify the status of the software download.

The MIB object docsDevSwAdminStatus defaults to ignoreProvisioningUpgrade after a modem has been upgraded using SNMP. This prevents a modem from upgrading via the configuration file the next time a bulk upgrade is performed. To restore the original value of allowProvisioningUpgrade, perform the following steps in this procedure.

15 Type the IP address of the cable modem under the Name or IP Address field.

16 Type **private** (or the proper Set Community name) in the Community field.

17 Highlight docsDevMIBObjects, then click **Down Tree**.

18 Highlight docsDevSoftware MIB, then click **Down Tree**.

19 From the MIB Values field, highlight **docsDevSwAdminStatus**.

20 From the SNMP Set Value field, type **2** (**allowProvisioningUpgrade**), then click **Set**.

Upgrading Software on All Cable Modems

The simplest way to update the software on all cable modems is to force cable modems to reset and specify a new software download image in the configuration file.

1 Modify the configuration file using the CMTS vendor's configuration file editor so that it specifies the new software download image filename.

2 Make sure that the configuration file includes the Software Upgrade TFTP Server Address where the new software download image is located.

3 Reset all cable modems on the CMTS by using the **clear cable modem all reset** command or by using SNMP to set the docsDevResetNow MIB object on all cable modems to True(1). This forces all modems to reset. The reset process forces the cable modems to reacquire the RF signal and reregister with the CMTS. The cable modems download the new configuration file, which specifies a new software download image. Because the name of the new image does not match the software image of the cable modems, all cable modems download this new image.

4 After the downloading process has started, you can monitor the process using the docsDevSwOperStatus MIB object. During the download, this object returns a value of inProgress(1) and the Test LED on the front panel of the cable modem blinks.

5 If downloading fails, the docsDevSwOperStatus MIB object returns a value of failed(4).

6 If downloading is successful, the cable modem automatically resets and the docsDevSwOperStatus MIB object returns a value of completeFromProvisioning(2).

7 The docsDevSwAdminStatus MIB object automatically resets itself to ignoreProvisioningUpgrade(3). If desired, set the docsDevSwAdminStatus MIB object to allowProvisioningUpgrade(2), to allow software updates via the configuration file.

8 Configuring Security

Management security can be implemented in a number of ways:

- Use the two Fast Ethernet ports to physically separate user data from management data or;
- Restrict access at each interface using the **management-access** specification or;
- Use ACLs to restrict access to/from the Cadant C3 at any sub-interface or;
- Use subscriber management filters to restrict access by CPE devices or;
- Use VLANs to separate user data from cable-modem and CMTS data or;
- Use the Cadant C3 cable sub-interface native VLAN and down-stream privacy capability to isolate user groups from one another.

The following sections discuss and explain each of these methods.

Physically Separating Data

The C3 has two physical FastEthernet interfaces, allowing C3 management to use a physically different interface to that used by subscriber traffic.

Bridge groups can be used to isolate CPE traffic from management traffic. The factory default C3 has two bridge groups pre-defined and allocated as follows:

```
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.0

fastethernet 0/0.0
bridge-group 0
no shutdown

cable 1/0.0
bridge-group 0
no shutdown

fastethernet 0/1.0
    bridge-group 1
    no shutdown

cable 1/0.1
    bridge-group 1
    encapsulation dot1q 1
    shutdown
```

In this configuration:

- Both modems and CPE are mapped to the cable 1/0.0 sub-interface
- Any broadcast traffic received at the cable sub-interface 1/0.0 is broadcast to the fastethernet 0/0.0 interface.

The CMTS management IP address can be assigned to either fastethernet 0/0.0 or 0/1.0.

> ***Note:*** You can assign the managment address to a cable sub-interface, but this is not recommended since shutting down the cable sub-interface also disables management access.

By adding the management IP address to fastethernet 0/1.0 and using the **management-access** specification, CMTS management can be isolated from the CPE and CM traffic in bridge group 0 as follows:

```
default cm-sub-interface cable 1/0.0
default CPE-sub-interface cable 1/0.0

fastethernet 0/0.0
```

```
bridge-group 0
no management-access

cable 1/0.0
bridge-group 0
no management-access

fastethernet 0/1.0
    bridge-group 1
    ip address 10.0.0.1 255.255.255.0
    management-access

cable 1/0.1
    bridge-group 1
no management-access
encapsulation dot1q 1 native
```

If required, CM traffic can be isolated from CPE traffic by reassigning the default interface for CM traffic as follows. Both modem and CMTS management traffic now use fastethernet 0/1.0:

```
default cm subinterface cable 1/0.1
default cpe subinterface cable 1/0.0

fastethernet 0/0.0
bridge-group 0
no management-access

cable 1/0.0
bridge-group 0
no management-access

fastethernet 0/1.0
    bridge-group 1
    ip address 10.0.0.1 255.255.255.0
    management-access

cable 1/0.1
    bridge-group 1
no management-access
encapsulation dot1q 1 native
```

The modem and CMTS traffic can be separated at this fastethernet interface by using the VLAN sub-interface capability of the C3.

- Once a fastethernet sub-interface is removed from a bridge group, this sub-interface is then assumed by the C3 to be the management interface for the C3.

- Another sub-interface is created and bridged to the modems on cable 1/0.1.
- One of the fastethernet 0/1.X sub-interfaces must have a VLAN tag—the following example shows the tagging being assigned to fastethernet 0/1.1:

```
default cm subinterface cable 1/0.1
default cpe subinterface cable 1/0.0

fastethernet 0/0.0
    ! for CPE traffic
bridge-group 0
no management-access

cable 1/0.0
    ! for CPE trafffic
bridge-group 0
no management-access

fastethernet 0/1.0
    ! for CMTS management
no bridge-group
    ip address 10.0.0.1 255.255.255.0
    management-access

fastethernet 0/1.1
    ! for modem traffic
    bridge-group 1
    encapsulation dot1q 11

cable 1/0.1
    ! for modems
    bridge-group 1
no management-access
encapsulation dot1q 1 native
```

Note: This example still falls within the boundaries of the basic software license abilities; namely up to 3 sub-interfaces per bridge group, up to 2 bridge groups, one VLAN tag per sub-interface, and one management-only sub-interface allowed.

As other examples in this chapter show, access by CPE devices to the management network can also be restricted by:

- ACL
- Subscriber management filters

Filtering Traffic

The C3 supports subscriber management filtering and access control list (ACL) based filtering. You can also configure filters in the modem itself—this option, although not part of a CMTS user manual, should not be overlooked. For example, if upstream multicast traffic is to be eliminated, it is better to block this traffic at the modem (modem configuration file specified) before being propagated upstream than to block at the CMTS where the upstream bandwidth is already used.

At this point it is worth asking what you want to do with such filtering.

Subscriber management filters are upstream/downstream and modem and CPE specific and:

- Are defined in the CMTS in groups of filters.
- The CMTS configuration can specify one of these filter groups as the default for all modems and attached CPE.
- The CMTS defaults can be overridden using the cable modem provisioning system; the defaults may be overridden using TLVs in a modem configuration file by the TLV referencing different filters (filters still defined in the CMTS).

If Subscriber management filters are never going to be manipulated in this manner, then you should consider using ACLs. ACL filters are sub-interface and direction specific, form part of a sub-interface specification and may be used on any sub-interface in the CMTS.

In summary:

- ACL:
  - — Sub-interface specific and can be used for filtering fastethernet traffic as well as cable traffic
  - — Static configuration
  - — More flexible filtering
- Subscriber management:
  - — Cable-modem and CPE specific
  - — CMTS default behavior can be specified
  - — Default behavior can be overridden by cable modem configuration file TLVs passed to CMTS during registration.

See also: “cable filter group” on page 6-69, “cable submgmt default filter-group” on page 6-82, “show cable filter” on page 6-29, “access-list” on page 6-66, “ip access-group” on page 6-113.

Working with Access Control Lists

This section describes the **access-list** syntax for each type of Access Control List (ACL) definition. Common uses for ACLs include:

- Preventing illegal access to services provided by the C3, such as Telnet, DHCP relay, and SNMP, from sources external to it, such as CMs, CPEs or other connected devices.
- Preventing access to service via the C3; that is, traffic passing through the C3 can also be subjected to ACL-based filtering. For example, ACLs could prevent access to certain TCP ports on CPEs to block external access to proxies and other services.

The C3 applies ACLs to all network traffic passing through the CMTS.

ACLs and ACEs

Access Control Lists (ACLs) are lists of Access Control Entries (ACEs) that are used to control network access to a resource.

Up to 30 ACLs may be defined; each ACL can contain up to 20 ACEs.

The ACL-number defines the type of ACL being created or referred to:

Number	Type
1-99	Standard IP
100-199	Extended IP
1300-1999	Standard IP (expanded range)
2000-2699	Extended IP(expanded range)

Multiple use of the **access-list** command—each using the same ACL-number but with different parameters—creates a new ACE for the ACL referred to by the ACL-number.

Implicit Deny All

One important point to note about ACLs is that there is an implicit “deny all” ACE at the end of each ACL.

- If an ACL consists of a series of ACEs and no match is made for any ACE, the packet is denied.
- If an ACL number is referred to or is assigned to an interface but no ACEs have been defined for this ACL, the implicit “deny all” ACE is *not* acted on.

An example of this command is as follows:

```
access-list 102 permit 6 any eq 23
```

This ACL allows TCP (protocol 102) based traffic from any source IP address with a TCP source port of 23 (Telnet) to pass through. All other packets are denied since they match the implicit "deny all" ACE. Another more complete example is as follows.

```
access-list 102 permit 6 192.168.250.0 0.0.0.255 eq 23 10.0.0.0 0.0.0.255 gt
1023
```

This ACL passes all TCP based traffic from any host in the 192.168.250.0/24 network with a TCP source port of 23 (Telnet) to a host within the 10.0.0.0/16 network with a TCP destination port of greater than 1023 to pass through.

Standard ACL Definition

Syntax: **[no] access-list {*ACL-number*} {permit | deny} {host *ipaddr* | *ipaddr wildcard* | any}**

Creates a standard ACL definition with the specified entry, or adds a new entry to an existing ACL. The parameters are:

ACL-number

The ACL identifier. Value: **1** to **99** or **1300** to **1399**. The C3 supports up to 30 ACLs, with each ACL containing up to 20 ACEs.

ipaddr

A single IP address, or (when specified with *wildcard*) the base address of a subnet.

wildcard

The inverted mask defining the limits of a subnet. For example, if the subnet contains 256 addresses, the wildcard is **0.0.0.255**.

any

Matches any IP address.

Extended IP Definitions

Syntax: **[no] access-list {*ACL-number*} {permit | deny} {*protocol*} {host *source* | *source source-wildcard* | any} {host *dest* | *dest dest-wildcard* | any} [icmp-type [icmp-code]] [precedence *precedence*] [tos *tos*] [dscp *dscp*]**

Creates an ACL definition with the specified entry, or adds a new entry to an existing ACL. The parameters are:

ACL-number

The ACL identifier. Value: **100** to **199** or **2000** to **2699**. The C3 supports up to 30 ACLs, with each ACL containing up to 20 ACEs.

protocol

The IP protocol type: **0** to **255**, or one of the following:

Keyword	Description
ahp	Authentication Header Protocol
eigrp	EIGRP routing protocol
esp	Encapsulation Security Protocol
gre	GRE tunneling
icmp	Internet Control Message Protocol
igp	IGP routing protocol
ip	any Internet protocol
ipinip	IP in IP tunneling
nos	KA9Q NOS compatible IP over IP tunneling
ospf	OSPF routing protocol
pcp	Payload Compression Protocol
pim	Protocol Independent Multicast
tcp	Transmission Control Protocol
udp	User Datagram Protocol

icmp-code

See “ICMP Definition” on page 8-10.

precedence

Matches the precedence bits of the IP header’s TOS field. Value: **0** to **7**, or one of the following:

Keyword	Description	Value
network	Match packets with network control precedence	7
internet	Match packets with internetwork control precedence	6
critical	Match packets with critical precedence	5
flash-override	Match packets with flash override precedence	4
flash	Match packets with flash precedence	3

Keyword	Description	Value
immediate	Match packets with immediate precedence	2
priority	Match packets with priority precedence	1
routine	Match packets with routine precedence	0

tos

Matches Type of Service (TOS) bits in the IP header's TOS field. Value: one of **0**, **2**, **4**, **8**, **16**, or one of the following:

Keyword	Description	Value
min-delay	Match packets with minimum delay TOS	8
max-throughput	Match packets with maximum throughput TOS	4
max-reliability	Match packets with maximum reliability TOS	2
min-monetary-cost	Match packets with minimum monetary cost TOS	1
normal	Match packets with normal TOS	0

dscp

The Differentiated Services Codepoint value: **0** to **63**, or one of the following:

Keyword	Description	Binary Value
af11	Match packets with AF11 dscp	001010
af12	Match packets with AF12 dscp	001100
af13	Match packets with AF13 dscp	001110
af21	Match packets with AF21 dscp	010010
af22	Match packets with AF22 dscp	010100
af23	Match packets with AF23 dscp	010110
af31	Match packets with AF31 dscp	011010
af32	Match packets with AF32 dscp	011100
af33	Match packets with AF33 dscp	011110
af41	Match packets with AF41 dscp	100010
af42	Match packets with AF42 dscp	100100
af43	Match packets with AF43 dscp	100110
cs1	Match packets with CS1 (precedence 1) dscp	001000

Keyword	Description	Binary Value
cs2	Match packets with CS2 (precedence 2) dscp	010000
cs3	Match packets with CS3 (precedence 3) dscp	011000
cs4	Match packets with CS4 (precedence 4) dscp	100000
cs5	Match packets with CS5 (precedence 5) dscp	101000
cs6	Match packets with CS6 (precedence 6) dscp	110000
cs7	Match packets with CS7 (precedence 7) dscp	111000
default	Match packets with default dscp	000000
ef	Match packets with EF dscp	101110

ICMP Definition

Syntax: **[no] access-list{*ACL-number*} {permit | deny} {icmp} {host *source* | *source source-wildcard* | any} {host *dest* | *dest dest-wildcard* | any} [*icmp-type* [*icmp-code*]] [fragment] [precedence *precedence*] [tos *tos*] [dscp *dscp*]**

Creates an ACL with the specified ICMP filter entry, or adds the specified ICMP filter entry to an existing ACL. The parameters are:

fragment

See “Fragment support” on page 8-16.

icmp-code

One of the following:

icmp-type	icmp-code	Equivalent CLI Keyword	Query	Error
0		echo-reply	X	

icmp-type	icmp-code	Equivalent CLI Keyword	Query	Error
3		destination-unreachable		
	0	net-unreachable		X
	1	host-unreachable		X
	2	protocol-unreachable		X
	3	port-unreachable		X
	4	fragment-needed-and-dont-fragment-was-set		X
	5	source-route-failed		X
	6	destination-network-unknown		X
	7	destination-host-unknown		X
	8	source-host-isolated (obsolete)		X
	9	communication-with-destination-network-is-admin-prohibited		X
	10	communication-with-destination-host-is-admin-prohibited		X
3	11	destination-network-unreachable-for-type-of-service		X
	12	destination-host-unreachable-for-type-of-service		X
	13	communication-admin-prohibited (by filtering)		X
	14	host-precedence-violation		X
	15	precedence-cutoff-in-effect		X
4		Source quench		X
5		redirect		
	0	redirect-datagram-for-the-network-or-subnet		X
	1	redirect-datagram-for-the-host		X
	2	redirect-datagram-for-the-type-of-service-and-network		X
	3	redirect-datagram-for-the-type-of-service-and-host		X
8		echo-request	X	

icmp-type	icmp-code	Equivalent CLI Keyword	Query	Error
9		router-advertisement	X	
	0	normal-router-advertisement	X	
	16	does-not-route-common-traffic	X	
10		router-selection	X	
11		time-exceeded		
	0	time-to-live exceeded-in-transit		X
	1	fragment-reassembly-time-exceeded		X
12		parameter-problem		
	0	pointer-indicates-the-error		X
	1	missing-a-required-option		X
	2	Bad-length		X
13		timestamp	X	
14		timestamp-reply	X	
15		information-request	X	
16		information-reply	X	
17		address-mask-request	X	
18		address-mask-reply	X	
30		traceroute	X	
31		datagram-conversion-error		X
32		mobile-host-redirect	X	
33		ipv6-where-are-you	X	
34		ipv6-I-am-here	X	
37		domain-name-request	X	
38		domain-name-reply	X	
39		skip	X	

icmp-type	icmp-code	Equivalent CLI Keyword	Query	Error
40		photuris		
	0	bad-spi		
	1	authentication-failed		
	2	decompression-failed		
	3	decryption-failed		
	4	need-authentication		
	5	need-authorisation		

Note that icmp-types **destination-unreachable**, **redirect**, **router-advertsiements**, **time-exceeded**, **parameter-problem**, and **photuris** have explicit code values associated with them. Other icmp-types have an implicit (not listed) code value of zero and thus no icmp-code option is expected at the CLI level.

TCP Definition

Syntax: **[no] access-list{*ACL-number*} {permit | deny} tcp {host *source* | *source source-wildcard* | any} [*oper port*] {host *dest* | *dest dest-wildcard* | any} [*oper port*] [*icmp-type* [*icmp-code*]] [fragment] [precedence *precedence*] [tos *tos*] [dscp *dscp*]**

Creates an ACL with the specified TCP filter entry, or adds the specified TCP filter entry to an existing ACL. The parameters are:

oper

Optional port specifier; one of **eq** (equal), **neq** (not equal), **lt** (less than), or **gt** (greater than).

port

The port number to match (using the defined operator): **0** to **65535**, or one of the following:

Keyword	Name	Port number
bgp	Border Gateway Protocol	179
chargen	Character generator	19
cmd	Remote commands (rcmd)	514
daytime	Daytime	13
discard	Discard	9
domain	Domain Name Service	53

Keyword	Name	Port number
echo	Echo	7
exec	Exec (rsh)	512
finger	Finger	79
ftp	File Transfer Protocol	21
ftp-data	FTP data connections (used infrequently)	20
gopher	Gopher	70
hostname	NIC hostname server	101
ident	Ident Protocol	113
irc	Internet Relay Chat	194
klogin	Kerberos login	543
kshell	Kerberos shell	544
login	Login (rlogin)	513
lpd	Printer service	515
nntp	Network News Transport Protocol	119
pim-auto-rp	PIM Auto-RP	496
pop2	Post Office Protocol v2	109
pop3	Post Office Protocol v3	110
smtp	Simple Mail Transport Protocol	25
sunrpc	Sun Remote Procedure Call	111
syslog	Syslog	514
tacacs	TAC Access Control System	49
talk	Talk	517
telnet	Telnet	23
time	Time	37
uucp	Unix-to-Unix Copy Program	540
whois	Nicname	43
www	World Wide Web (HTTP)	80

tcpflags

Matches TCP header flags. Value: A six-bit value, **0** to **63**, where:

Bit	Name
5	urgent
4	ack
3	push
2	reset
1	sin
0	fin

UDP Definition

Syntax: **[no] access-list{*ACL-number*} {permit | deny} udp {host *source* | *source source-wildcard* | any} [*oper port*] {host *dest* | *dest dest-wildcard* | any} [*oper port*] [*icmp-type* [*icmp-code*]] [fragment] [precedence *precedence*] [tos *tos*] [dscp *dscp*]**

Creates an ACL with the specified UDP filter entry, or adds the specified UDP filter entry to an existing ACL. The parameters are:

oper

See “TCP Definition” on page 8-13.

port

The port number to match (using the defined operator): **0** to **65535**, or one of the following:

Keyword	Name	Port number
biff	Biff (mail notification, comsat)	512
bootpc	Bootstrap Protocol (BOOTP) client	68
bootps	Bootstrap Protocol (BOOTP) server	67
discard	Discard	9
dnsix	DNSIX security protocol auditing	195
domain	Domain Name Service (DNS)	53
echo	Echo	7
isakmp	Internet Security Association and Key Management Protocol	500
mobile-ip	Mobile IP registration	434
nameserver	IEN116 name service (obsolete)	42

Keyword	Name	Port number
netbios-dgm	NetBios datagram service	138
netbios-ns	NetBios name service	137
netbios-ss	NetBios session service	139
ntp	Network Time Protocol	123
pim-auto-rp	PIM Auto-RP	496
rip	Routing Information Protocol (router, in.routed)	520
snmp	Simple Network Management Protocol	161
snmptrap	SNMP Traps	162
sunrpc	Sun Remote Procedure Call	111
syslog	System Logger	514
tacacs	TAC Access Control System	49
talk	Talk	517
tftp	Trivial File Transfer Protocol	69
time	Time	37
who	Who Service (rwho)	513
xdmcp	X Display Manager Control Protocol	

All Other Protocols

Syntax: **[no] access-list {*ACL-number*} {permit | deny} {*protocol*} {host *source* | *source source-wildcard* | any} [*oper port*] {host *dest* | *dest dest-wildcard* | any} [*oper port*] [*icmp-type* [*icmp-code*]] [fragment] [precedence *precedence*] [tos *tos*] [dscp *dscp*]**

Creates an ACL with the specified filter entry, or adds the specified filter entry to an existing ACL.

The [no] Option

Use the **no** option to remove an ACE from a ACL without having to re-enter the complete ACL.

Fragment support

Full support of the fragment option is provided. Use this option to prevent attacks on hosts as detailed by RFC 1858. However, using this option restricts access to resources by non-fragment flows only.

The first packet of a TCP segment contains the IP header (Layer 3) and the TCP header (layer 4). This fragment is an “initial fragment.” Subsequent IP packets (fragments) of this segment only have a layer 3 header (no TCP header). Such fragments are “non-initial fragments.”

If a TCP segment is completely contained in the first IP Datagram then this is a “non-fragment” packet.

With regard to defining ACL filters, blocking initial fragments is often all that is required as the remaining packets cannot be re-assembled; that is, all packets with an offset greater than zero traditionally are allowed to pass through ACL filters. But this type of processing can allow both an overlapping fragment attack and a tiny fragment attack on the host as detailed in RFC1858. Thus, the C3 must also be able to deny non-initial fragments.

Where a data flow to port 80 on a host is to be protected, an ACL such as ACL 100 (see below) may be created. This ACL only tests for initial fragments.

When an ACL such as ACL102 (see below) is created, non-initial fragments (containing no layer 4 header) match the layer 3 part of the first ACE. As there is no Layer 4 information in the packet, no layer 4 information is tested. This packet is a non-initial fragment, so the fragment option also matches. Thus, all ACE filter options that can be matched are matched and the packet is denied.

In the case where an initial or non fragment hits this first ACE, the layer 3 filter matches, the layer 4 filter (port number) matches but this packet is an initial (or non-) fragment so the last filter—the fragment option— fails and the packet will be passed to the next ACE in the ACL.

Example:

```
access-list 100 permit tcp any host 192.168.253.65 eq 80
access-list 100 deny ip any any
```

This filter, applied to the C3 as an incoming filter, is designed to permit only HTTP (port 80) to the host 192.168.253.65. But is this true? A non-initial fragments HTTP packet (a packet with an incomplete layer 4 header) can also pass to the specified hostm opening the host to an overlapping fragment or a tiny fragment attack.

```
access-list 102 deny ip any host 192.168.253.65 fragments
access-list 102 permit tcp any host 192.168.253.65 eq 80
access-list 102 deny ip any any
```

If filter 102 is applied, all non-initial fragments are denied and only non-fragmented HTTP data flows are permitted through to the specified host.

Using an ACL

Defining an ACL does not actually apply the ACL for use.

Use the **ip access-group** command to associate an ACL with inbound or outbound traffic on a specific interface or sub-interface.

It is not necessary, nor is it recommended, to apply an ACL to block protocols in a symmetrical manner. For example, to block PING access to an interface on the C3, it is only necessary to block either the ICMP echo or the ICMP reply—blocking either will block ping—so assigning only an inbound ACL is sufficient.

> ***Note:*** ACLs can be associated to interfaces before the ACL is defined. Undefined ACLs assigned to an active interface using the **ip access-group** command (ACL number assigned but the actual ACL is not defined) are *not* ignored by the interface. Undefined ACLs on active interfaces still contain the implicit "deny all" ACE resulting in the dropping of all packets seen at that interface.

Example:

```
fastethernet 0/1.1
  ip access-group 101 in
! ACL 101 has not been defined
```

Since ACL 101 has not been defined, the C3 does not permit any packets on that interface (and sub-interface) for the direction that the ACL was configured on, in the above case the input direction.

The **ip access-group** command takes the following format, when configuring an interface:

access-group {ACL-number} {in | out}

An example of the command is as follows (note that the command only applies when configuring an interface):

```
C3>enable
C3#config t
(config-t)>interface fastethernet 0/0
(fastethernet 0/0)> ip access-group 102 in
(fastethernet 0/0)> ip access-group 103 out
(fastethernet 0/0)> ^z
```

This configuration associates ACL number 102 to incoming traffic on the fastethernet 0/0 interface, and ACL number 103 to outgoing traffic.

Example

The network must support the following features:

- CPEs can be allocated to a number of different subnets.
- No CPE with a static address should be useable on any subnet other than the assigned subnet.
- No CPE should have access to modem subnets.

One solution to this problem involves a mixture of ACL and subscriber management based filtering and provides a good example of the differences in these filtering techniques.

Note that it is possible to solve this problem using bridge groups, sub-interfaces, and ACLs per sub-interface; but the point of this example is to show the use of ACL and subscriber management filtering.

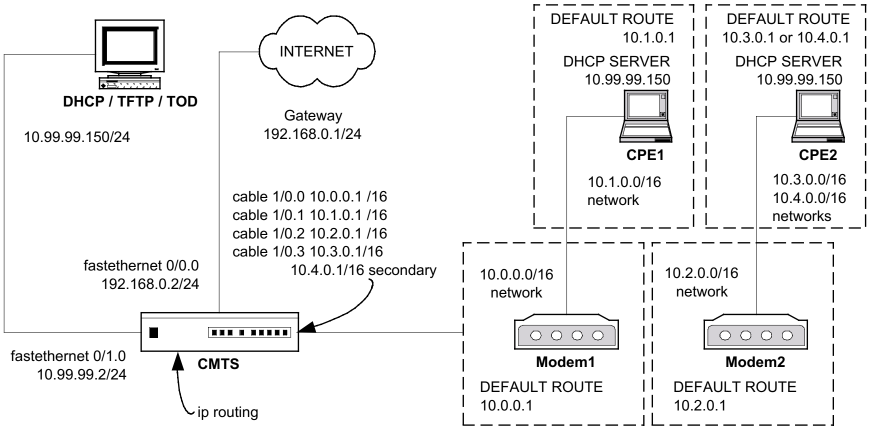
Blocking CPE access to modems is relatively straight forward. All the CPE subnets are known and are static. Use ACLs to drop all packets from the CPE subnets destined for modem subnets. One ACL could be used on all CPE sub-interfaces.

> ***Note:*** If some CPEs must have access to modems (MSO technicians working from home) then the use of ACLs is still appropriate as these modems and hence attached CPE can be allocated to a known sub-interface by the provisioning system, a sub-interface that does not have so restrictive an ACL specification. Blocking a manually set CPE static IP address allocation providing access to “illegal” CPE subnets is not a static situation suitable for ACL application. The assigned subnet may be one of many subnets defined for a cable sub-interface. An ACL can protect against attempts to spoof an address outside the defined subnets for this sub-interface, but cannot be used to isolate a CPE to one subnet of the many in this situation. The “valid” subnet for this CPE is not known in advance by the CMTS. All the possible CPE subnets are known, but which one is used by this CPE? An ACL cannot be specified and is thus not appropriate in this case.

It is not until the modem is provisioned and allocated to an IP address space that attached CPE are allocated to an IP address space. The use of submgmt filters in this case allows one of many predefined filters in the CMTS to now be applied based on the modem provisioning. This filter-group would act on CPE packets and accept any packet with a source IP address in a subnet and drop all other packets. The CMTS can have pre-defined in it all such possible filters (one per CPE subnet). The correct filter-group number for the desired valid CPE subnet is then referenced in the modem configuration file and passed to the CMTS during modem registration; i.e. after the modem registers with the CMTS, this filter-group number will be assigned to any CPE attached to this modem. The result being even if a static IP address is given to a CPE, it will not provide any network access unless within the correct subnet.

Sample network

The following is a simplified network diagram for this example.



Sample ACL definition

The following commands configure ACLs to provide the functionality described above.

```
! Requirement:
!   Block any CPE from accessing the cable modem address space.
!   Block CPE access to the DHCP server address space
!   except for DHCP
!   Block CPE from access to CMTS 192.168.0.2 port
configure terminal
! deny cpe on on cable 1/0.1 access to any modem subnets
access-list 101 deny ip 10.1.0.0 0.0.255.255 10.0.0.0 0.0.255.255
access-list 101 deny ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
! deny cpe on cable 1/0.1 ip access to 10.99.99.0 network
access-list 101 deny ip 10.1.0.0 0.0.255.255 10.99.99.0 0.0.0.255
! deny cpe on cable 1/0.1 ip access to 192.168.0.2
access-list 101 deny ip 10.1.0.0 0.0.255.255 192.168.0.2 0.0.0.0
! permit cpe on cable 1/0.1 dhcp access to 10.99.99.0 network
access-list 101 permit udp 10.1.0.0 0.0.255.255 10.99.99.150 0.0.0.0 eq bootpc
! permit all remaining ip
! remember that the last ACE is always an implicit deny all
access-list 101 permit ip any any
!

! deny cpe on cable 1/0.3 access to any modem subnets
access-list 103 deny ip 10.3.0.1 0.0.255.255 10.0.0.0 0.0.255.255
access-list 103 deny ip 10.3.0.1 0.0.255.255 10.2.0.0 0.0.255.255
access-list 103 deny ip 10.4.0.1 0.0.255.255 10.0.0.0 0.0.255.255
access-list 103 deny ip 10.4.0.1 0.0.255.255 10.2.0.0 0.0.255.255
! deny cpe on cable 1/0.3 access to 10.99.99.0 network
access-list 103 deny ip 10.1.0.0 0.0.255.255 10.99.99.0 0.0.0.255
! deny cpe on cable 1/0.3 ip access to 192.168.0.2
```

```
access-list 103 deny ip 10.3.0.0 0.0.255.255 192.168.0.2 0.0.0.0
access-list 103 deny ip 10.4.0.0 0.0.255.255 192.168.0.2 0.0.0.0
! permit cpe on cable 1/0.3 dhcp access to 10.99.99.0 network
access-list 103 permit udp 10.3.0.0 0.0.255.255 10.99.99.150 0.0.0.0 eq bootpc
access-list 103 permit udp 10.4.0.0 0.0.255.255 10.99.99.150 0.0.0.0 eq bootpc
! permit all remaining ip
! remember that the last ACE is always an implicit deny all
access-list 103 permit ip any any
!
interface cable 1/0.1
ip access-group 101 in
interface cable 1/0.3
ip access-group 103 in
exit
exit
```

Sample subscriber management filter definition

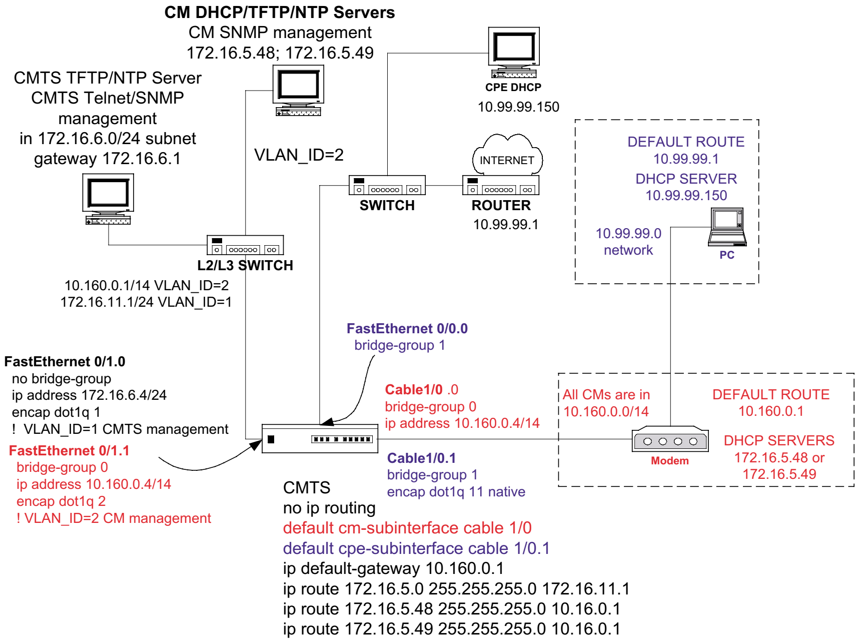
The following commands define subscriber management filters to provide the functionality described above.

```
! Requirement: define filters that can be referenced from modem
! configuration files that restrict CPE source address to a
! defined subnet.
! Assign default CMTS submgmt filters to block all
! IP based CPE access for the default subscriber management filters
!
configure terminal
!
! define filter group for CPE network 10.1.0.0
cable filter group 1 index 1
cable filter group 1 index 1 src-ip 10.1.0.0
cable filter group 1 index 1 src-mask 255.255.0.0
cable filter group 1 index 1 dest-ip 0.0.0.0
cable filter group 1 index 1 dest-mask 0.0.0.0
cable filter group 1 index 1 ip-proto ALL
cable filter group 1 index 1 ip-tos 0x0 0x0
cable filter group 1 index 1 match-action accept
cable filter group 1 index 1 status activate
cable filter group 1 index 1 src-port all
cable filter group 1 index 1 dest-port all
cable filter group 1 index 1 tcp-flags 0x0 0x0


!
! define a default action for this filter group ie drop all
!
cable filter group 1 index 2
cable filter group 1 index 2 src-ip 0.0.0.0
cable filter group 1 index 2 src-mask 0.0.0.0
cable filter group 1 index 2 dest-ip 0.0.0.0
cable filter group 1 index 2 dest-mask 0.0.0.0
cable filter group 1 index 2 ip-proto ALL
cable filter group 1 index 2 ip-tos 0x0 0x0
cable filter group 1 index 2 match-action drop
cable filter group 1 index 2 status activate

!
! define filter group for CPE network 10.3.0.0
!
cable filter group 3 index 1
cable filter group 3 index 1 src-ip 10.3.0.0
cable filter group 3 index 1 src-mask 255.255.0.0
cable filter group 3 index 1 dest-ip 0.0.0.0
cable filter group 3 index 1 dest-mask 0.0.0.0
cable filter group 3 index 1 ip-proto ALL
cable filter group 3 index 1 ip-tos 0x0 0x0
cable filter group 3 index 1 match-action accept
cable filter group 3 index 1 status activate
cable filter group 3 index 1 src-port all
cable filter group 3 index 1 dest-port all
cable filter group 3 index 1 tcp-flags 0x0 0x0


!
! define a default action for this filter group ie drop all
!
cable filter group 3 index 2
cable filter group 3 index 2 src-ip 0.0.0.0
cable filter group 3 index 2 src-mask 0.0.0.0
cable filter group 3 index 2 dest-ip 0.0.0.0
cable filter group 3 index 2 dest-mask 0.0.0.0
cable filter group 3 index 2 ip-proto ALL
cable filter group 3 index 2 ip-tos 0x0 0x0
cable filter group 3 index 2 match-action drop
cable filter group 3 index 2 status activate

!
! define filter group for CPE network 10.4.0.0
!
cable filter group 4 index 1
cable filter group 4 index 1 src-ip 10.4.0.0
cable filter group 4 index 1 src-mask 255.255.0.0
cable filter group 4 index 1 dest-ip 0.0.0.0
cable filter group 4 index 1 dest-mask 0.0.0.0
cable filter group 4 index 1 ip-proto ALL
cable filter group 4 index 1 ip-tos 0x0 0x0
cable filter group 4 index 1 match-action accept
cable filter group 4 index 1 status activate
cable filter group 4 index 1 src-port all
cable filter group 4 index 1 dest-port all
cable filter group 4 index 1 tcp-flags 0x0 0x0

!
! define a default action for this filter group ie drop all
!
cable filter group 4 index 2
cable filter group 4 index 2 src-ip 0.0.0.0
cable filter group 4 index 2 src-mask 0.0.0.0
cable filter group 4 index 2 dest-ip 0.0.0.0
cable filter group 4 index 2 dest-mask 0.0.0.0
cable filter group 4 index 2 ip-proto ALL
cable filter group 4 index 2 ip-tos 0x0 0x0
cable filter group 4 index 2 match-action drop
cable filter group 4 index 2 status activate

!
! define a default filter group to block all access from CPE
! so if mistake made with modem config file no danger of illegal
! access.
!
! Note this will block all CPE access if the modem config file
! does not call the correct filter-group id
!
cable filter group 99 index 1
cable filter group 99 index 1 src-ip 0.0.0.0
cable filter group 99 index 1 src-mask 0.0.0.0
cable filter group 99 index 1 dest-ip 0.0.0.0
cable filter group 99 index 1 dest-mask 0.0.0.0
cable filter group 99 index 1 ip-proto ALL
cable filter group 99 index 1 ip-tos 0x0 0x0
cable filter group 99 index 1 match-action drop
cable filter group 99 index 1 status activate
cable filter group 99 index 1 src-port all
cable filter group 99 index 1 dest-port all
cable filter group 99 index 1 tcp-flags 0x0 0x0
!

! activate filters
cable filter
! turn on subscriber managment in the CMTS
cable submgmt
! up to 16 cpe addresses per modem can be learned by the CMTS
cable submgmt default max-cpe 16
! let the cmts learn the attached cpe ip addres up to the maximum (16)
cable submgmt default learnable
! filter cpe traffic based on learned cpe ip address up to the maximum (16)
cable submgmt cpe ip filtering
! activate the defaults defined here for all modems and attached cpe
cable submgmt default active

! Assign default filters
cable submgmt default filter-group cm upstream 99
cable submgmt default filter-group cm downstream 99
cable submgmt default filter-group cpe upstream 99
cable submgmt default filter-group cpe downstream 99
!
! Now all set for a modem config file submgmt TLV to reference
! filter group 1 for CPE in network 10.1.0.0
! filter group 3 for CPE in network 10.3.0.0
! filter group 4 for CPE in network 10.4.0.0
```

```
!
exit
```

Using Simple VLANS to Isolate Modem and CMTS Traffic

Previous version of the C3 firmware supported the cable vpn command. This command is now redundant due to the extensive enhancements to the C3 VLAN and VPN capabilities. This section shows how to configure a C3 for the equivalent function of the old cable vpn command using the base C3 software license.



In the above diagram, all broadcast modem traffic is mapped to the cable 1/0.0 sub-interface by the **default cm sub-interface** specification, and thus to bridge group 0. This bridge group bridges traffic to fastethernet 0/1.1 and is thus VLAN encoded with tag 2 and sent to the L2/L3 switch then to the CM DHCP servers.

Modem discover broadcast, however, is unicast by the DHCP Relay function to both 172.16.5.48 and 172.16.5.49. This subnet is not directly connected to the C3, so is routed using the defined host routes to the L2/L3 switch at 10.160.0.1. Again, modem Renew is directed to either 172.16.5.48 or 172.16.5.49, depending on which answered the original DHCP. Again these packets will be routed using the host routes.

All CPE traffic is mapped to cable 1/0.1 (on bridge group 1) and bridged to the fastethernet 0/0.0 sub-interface. CPE devices have no specified DHCP relay, so the C3 broadcasts DHCP from the fastethernet 0/0.0 sub-interface to the DHCP server. DHCP relay could be activated if required, in which case the cable 1/0.1 sub-interface would need an IP address—preferably in the subnet required for the CPE devices.

Fastethernet 0/1.0 is not a member of any bridge group and will thus be assumed by the CMTS to be a CMTS management interface only. Traffic from the CMTS to the 172.16.5.0 network is destined for a network not connected to the C3. To assist, a static route is added for this network via 172.16.11.1

The following is a sample configuration for the diagram above.

```
! if the following is to be pasted to the command line then paste from
! privilege mode and paste over a factory default configuration.
! Restore factory default using
!    write erase
!    reload
! then select do not save configuration and select yes to restart
!------------ start script ---------------------
configure terminal
no ip routing
default cm-subinterface cable 1/0.0
default cpe-subinterface cabel 1/0.1
!
interface fastethernet 0/0.0
! for all CPE traffic
! no ip address required
bridge-group 1
no shutdown
no management-access
!
interface fastethernet 0/1.0
! for CMTS management
! remove the factory default assignment
no bridge-group
! set management IP address
ip address 172.16.11.4 255.255.255.0
management-access
encapsulation dot1q 1
no shutdown
exit

!
interface fastethernet 0/1.1
! for modem traffic
bridge-group 0
ip address 10.160.0.4 255.252.0.0
no management-access
no shutdown
encapsulation dot1q 2
!
interface cable 1/0.0
! for modem traffic
```

```
bridge group 0
! get basic rf going
no shutdown
no cable upstream 0 shutdown
ip address 10.160.0.4 255.252.0.0
no management-access
ip dhcp relay
ip dhcp relay information option
cable dhcp-giaddr primary
cable helper-address 172.16.5.48
cable helper-address 172.16.5.49
exit

!
cable 1/0.1
! for CPE traffic
bridge-group 1
encapsulation dot1q 11 native
no ip dhcp relay
exit
!
! set the bridge mode default gateway
ip default-gateway 10.160.0.1
!
! route all traffic to network 172.16.5.0 to
! fa 0/1.1 and thus VLAN tag 1 for CMTS management
ip route 172.16.5.0 255.255.255.0 172.16.11.1
!
! add specific host routes for DHCP servers as they are on the same
! subnet as the CMTS traffic but a different VLAN
! ie force modem traffic to fa 0/1.1 and thus VLAN tag 2 for CM management
ip route 172.16.5.48 255.255.255.0 10.160.0.1
ip route 172.16.5.49 255.255.255.0 10.160.0.1
exit
!--------------- end script --------------------
```

Encrypting Native VLANS

Access to the C3 itself may be secured using techniques defined in this chapter, but the C3 may also be configured to prevent:

- IP address spoofing of modems by CPE devices
- Spoofing of IP addresses by CPE devices to access the management system
- Spoofing of 802.1Q VLAN tags by CPE devices

The cable sub-interfaces on the C3 can be used to:

- restrict layer 2 traffic to the attached bridge-group;
- restrict access to defined IP subnets and
- restrict access to defined VLANS for devices allocated to cable sub-interfaces.

Such restrictions are enforced by placing CPE devices in a native VLAN using either VSE encoding or using the **map-cpes** command. Both commands map all CPE traffic to defined cable sub-interfaces and thus force CPE traffic to obey the specifications of the this sub-interface.

Both options also allow the CPE assigned to a cable sub-interface and hence native VLAN to be placed in private downstream broadcast domains by using separately keyed downstream encryption for each native VLAN using the **encapsulation dot1q xx encrypted-multicast**command.

Example:

```
conf t
ip routing
cable 1/0.1
no bridge-group
ip address 10.1.0.1 255.255.0.0
ip address 10.2.0.1 255.255.0.0 secondary
ip source verify subif
exit
exit
```

In IP routing mode, this restricts access by CPE allocated to this sub-interface to the stated subnets only.

Example (routing case):

```
conf t
ip routing
cable 1/0.1
```

```
    no bridge-group
ip address 10.1.0.1 255.255.0.0
encapsulation dot1q 5
exit
exit
```

Example (hybrid case):

```
conf t
ip routing
cable 1/0.1
    bridge-group 0
ip address 10.1.0.1 255.255.0.0
encapsulation dot1q 5
exit
exit
```

Example (bridging case):

```
conf t
no ip routing
cable 1/0.1
    bridge-group 0
encapsulation dot1q 5
exit
exit
```

This restricts access by CPE allocated to this sub-interface to those CPE that generate 802.1Q encoded data and with a vlan tag of 5.

In the above cases, the CPE incoming data is allocated by the Cadant C3 to the specified cable sub-interfaces using 802.1Q tags generated by the CPE devices.

Example:

In the following sample configuration:

- All modems use the cable 1/0.0 sub-interface for initial DHCP.
- Regardless of the cable sub-interface used by a modem, VSE encoding in a modem configuration file modem directs attached CPE to either the cable 1/0.11 or the cable 1/0.13 sub-interfaces and hence subject to the restrictions imposed by these sub-interface's specifications.
- The default CPE sub-interface has been specified as cable 1/ 0.13.
- In the case of CPE traffic allocated to cable 1/0.11, incoming frames may be layer 2—they are bridged using bridge group 1.
- In the case of CPE traffic allocation to cable 1/0.13, only layer 3 traffic is accepted (non bridging sub-interface) and CPE DHCP

is directed to only the DHCP server at 10.0.0.1; CPE source IP addresses must belong to subnet 10.11.0.0/16 or be dropped.

```
conf t
ip routing
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.13
bridge 1
!
cable 1/0.0
! for modem DHCP only
ip address 10.99.99.1
ip dhcp relay
cable helper-address 10.0.0.1 host
cable dhcp-giaddr primary

cable 1/0.1
! for modems once allocated an IP address
ip address 10.99.98.1

cable 1/0.11
! for cpe layer 2 forwarding
! for CPE traffic via modem with VSE tag = 11
encapsulation dot1q 11 native
bridge-group 1

cable 1/0.13
! for cpe layer 3 forwarding
! for CPE traffic via modem with VSE tag = 13
no bridge-group
ip address 10.11.0.1 255.255.0.0
ip dhcp relay
cable helper-address 10.0.0.1 host
cable dhcp-giaddr primary
ip source verify subif
encapsulation dot1q 13 native
exit
exit
```

Example:

Modems can be mapped by source IP to other cable sub-interfaces. In the following example if the provisioning system allocated the modem to subnet 10.99.98.0, modem traffic will be allocated the cable 1/0.1 sub-interface.

The cable sub-interface cable 1/0.1 contains a map-cpes specification.

The map-cpes specification under this sub-interface directs attached CPE to the cable 1/0.11 sub-interface and hence subject to the restrictions imposed by these sub-interface’s specifications.

In this case, **ip source verify subif** is specified and thus CPE source IP address must belong to the 10.11.0.0/24 subnet or be dropped. ie CPE IP address cannot belong to another subnet.

```
conf t
ip routing
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.2
cable 1/0.0
! for modem DHCP only
no bridge-group
ip address 10.99.99.1
ip dhcp relay
cable helper-address 10.0.0.1 cable-modem
cable dhcp-giaddr primary

cable 1/0.1
! for modems once allocated an IP address
no bridge-group
encapsulation dot1q 1 native
ip address 10.99.98.1
map-cpes cable 1/0.11

cable 1/0.2
! for unprovisoned cpe
no bridge-group
ip address 10.1.0.1 255.255.255.0
ip source-verify subif
encapsulation dot1q 11 native
ip dhcp relay
cable helper-address 10.0.0.1 host
cable dhcp-giaddr primary

cable 1/0.11
! for cpe IP forwarding
no bridge-group
encapsulation dot1q 11 native
encapsulation dot1q 11 encrypted-multicast
ip address 10.11.0.1 255.255.255.0
ip source-verify subif
ip dhcp relay
cable helper-address 10.0.0.1 host
cable dhcp-giaddr primary
```

Selective use of cable sub-interfaces can define with tight limits the address space and layer 2/3 capabilities of CPE devices attached to modems.

9 Service Procedures

The procedures in this chapter cover basic maintenance and upgrade tasks.

Removing Power for Servicing

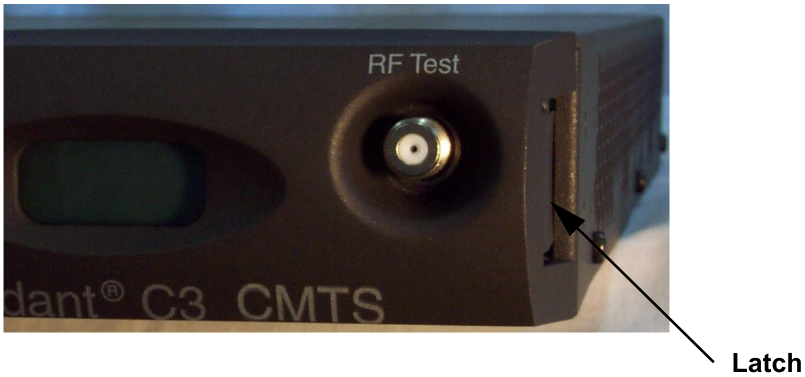
To disconnect power from the C3 for servicing, remove both power leads (AC and DC) from the rear of the chassis.

Front Panel Removal and Replacement

Removing the face plate can be done during normal system operation without any adverse impact.

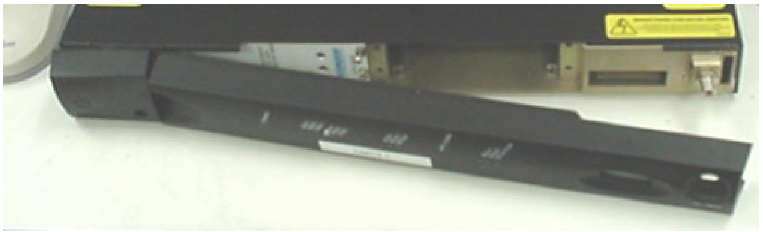
Action

1 Locate the indentation on the right side of the CMTS front panel.



2 Press the indentation to release the latch and then pull the right side of the faceplate away from the CMTS.

3 To reinstall the faceplate, place the left edge of the faceplate against the front of the fan tray so that the faceplate is at a 45 degree angle to the front of the CMTS. See the following photo.



4 Push the right side of the faceplate back towards the front of the CMTS slowly so that the edge connector on the rear of the faceplate mates properly with the connector on the front of the CMTS. Press the right side of the face plate in firmly to latch it to the CMTS.

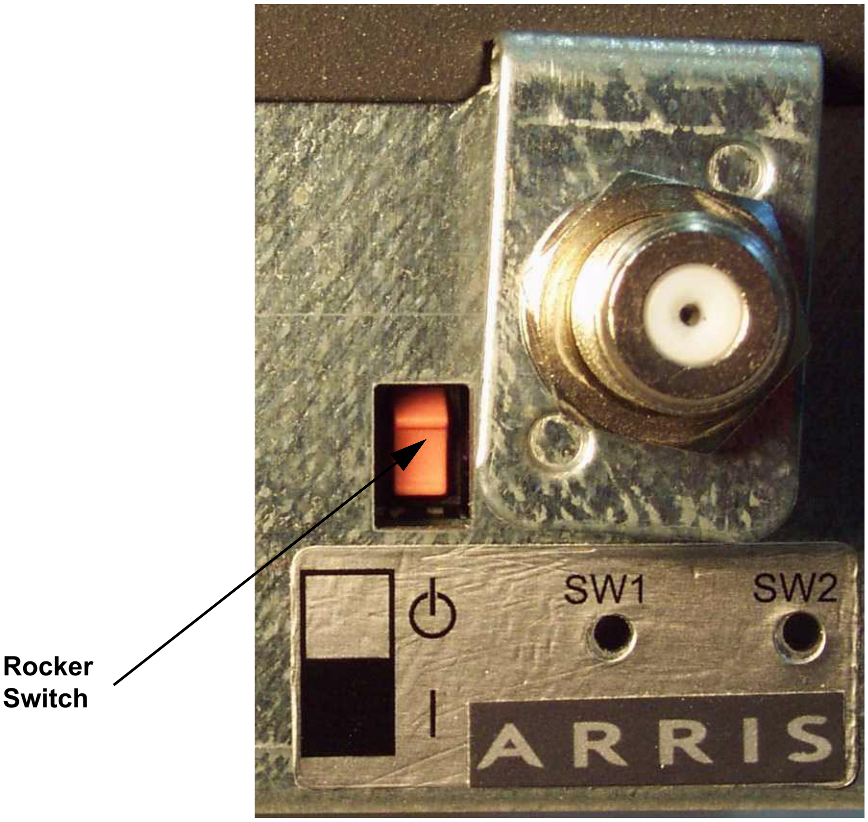
Resetting the Power Supplies

If a power supply shuts down for thermal reasons, the “F” Amber LED on the front of the power supply lights up. Use this procedure to reset the power supplies.

Action

1 Correct the thermal condition.

2 Reset the power supply by pushing the rocker switch near the RF test port up then press the rocker switch down to restart. The following figure shows the rocker switch in the RUN condition.



Note: Pressing the rocker switch up on a running CMTS shuts down the CMTS after copying the running configuration to the startup configuration. (Toggling the rocker switch again has no effect until the CMTS is fully booted again).

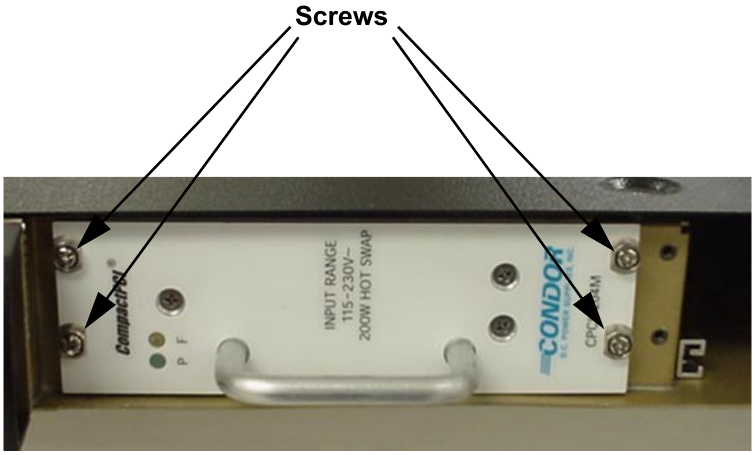
Replacing a Power Supply

The C3 CMTS can have two fully redundant power supplies. You can replace one supply without powering down the CMTS.

> ***Note:*** If only one power supply is installed and active, the CMTS shuts down once the power supply has been removed.

Diagram

Refer to the following photo while performing this procedure.



Action

1 Remove the front panel as described in “Front Panel Removal and Replacement” on page 9-2.

2 Loosen the four screws at the corners of the power supply.

3 Pull the supply towards the front of the CMTS using the silver handle.

The power supply slides out of the chassis.

4 Line up the replacement power supply with the slot, then push the power supply firmly into the slot.

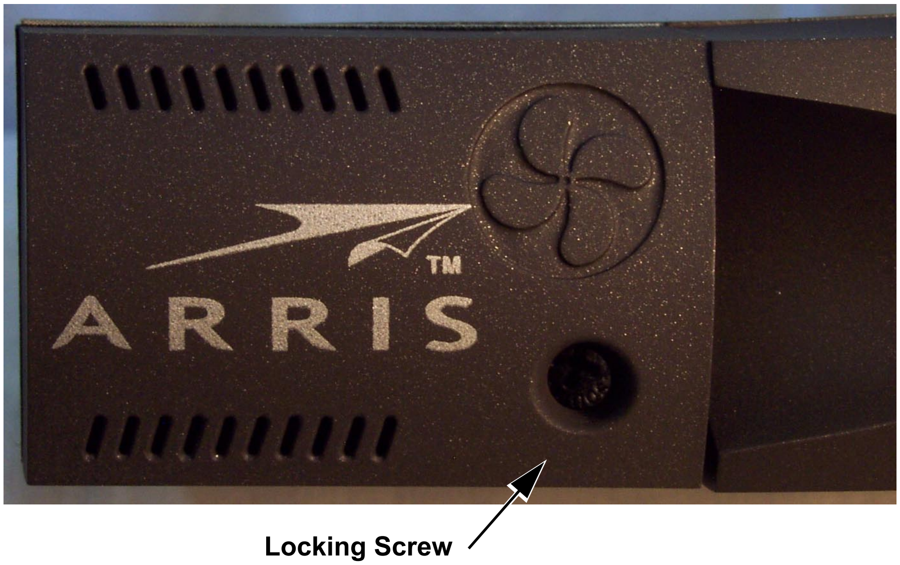
5 Use the four screws fitted to the new supply to secure the replacement power supply.

Fan Tray Replacement

You can replace the fan tray while the ARRIS Cadant C3 is running, as long as you finish inserting the replacement tray within 60 seconds. Beyond that time, the C3 CMTS starts to shut down as the monitored internal temperature rises.

Diagram

Refer to the following diagram for the location of the fan tray.



Action

Follow these steps to replace the fan tray.

1. Loosen the Phillips screw located in the front of the fan tray by turning the screw counter-clockwise. The screw rotates 90 degrees to unlock the fan tray; it does not remove completely.

2. Insert your finger behind the ARRIS logo and pull the fan tray out towards the front of the C3.

3. Insert the new fan tray into the opening, and secure it with the locking screw.

Replacing the Battery

The expected lifetime of the C3 CMTS battery is 10 years. This is an average expectancy and the actual battery lifetime may be shorter or longer.

Requirements

Replacing the battery requires a complete shutdown of the C3 CMTS.

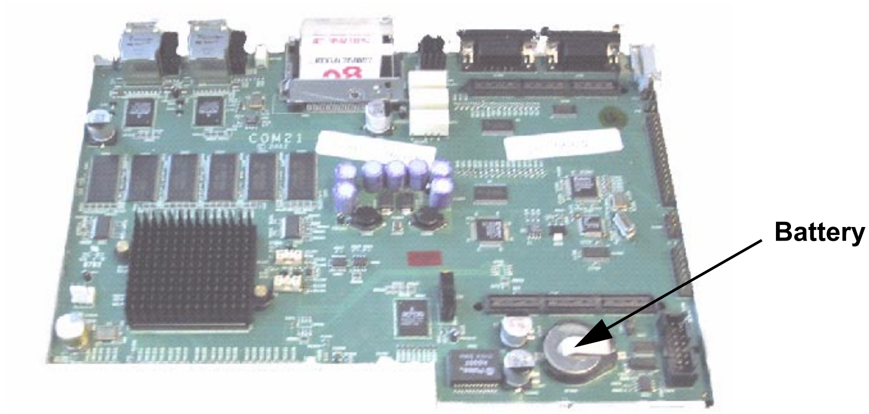
DANGER
Risk of injury from battery explosion
Risk of explosion if the battery is replaced by an incorrect type. Dispose of used batteries according to the manufacturer's instructions.

Battery type is CR3020 lithium.

Use anti-static precautions such as a wrist grounding strap grounded to a grounded work area when handling the CMTS CPU card.

Diagram

The following diagram shows the location of the battery on the CPU card.

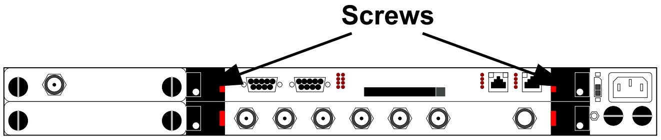


Action

1 Power down the CMTS by removing all power leads from the rear sockets.

2 Remove the CPU card from the CMTS chassis as follows:

a Loosen the two Phillips screws securing the CPU card to the chassis. The screws run through the black pull tabs on each end of the card.



b Push the red tabs towards the outer edge of the unit. The black latches will click when they have been released. Gently push the black latches towards the outer edge of the unit to release the card.

c Grasp the CPU by the black tabs on either end of the card and slide the card out of the chassis

3 Gently lift the spring metal contact over the battery and lift the battery from its holder. You may need to use a small screwdriver to gently pry the battery out of the holder.

4 Insert the new battery in the holder.

5 Replace the CPU card into the chassis:

a Line up the CPU card with the guides inside the chassis, and slide the card into the chassis.

b Push the card into the chassis until the latches click into place. Secure the card using the Phillips screws.

6 Replace the power connections.

Replacing the RF Card

The C3 may be shipped with 2, 4, or 6 upstreams.

Requirements

Contact your ARRIS representative to obtain a new upstream card.

Replacing the upstream card requires a complete shutdown of the C3 CMTS.

Use anti-static precautions such as a wrist grounding strap grounded to a grounded work area when handling the upstream card.

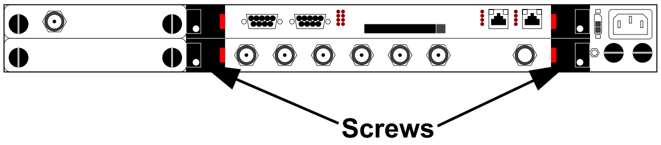
Action

1 Power down the CMTS by removing all power leads from the rear sockets.

2 Disconnect the upstream RF cables from the CMTS. Label the RF cables, if necessary, to prevent misconnection after replacing the upstream card.

3 Remove the upstream card from the CMTS chassis as follows:

   a Loosen the two Phillips screws securing the upstream card to the chassis. The screws run through the black pull tabs on each end of the card.



   b Push the red tabs towards the outer edge of the unit. The black latches will click when they have been released. Gently push the black latches towards the outer edge of the unit to release the card.

   c Grasp the upstream card by the black tabs on either end of the card and slide the card out of the chassis.

4 Install the new upstream card into the chassis:

   a Line up the upstream card with the guides inside the chassis, and slide the card into the chassis.

- **b** Push the card into the chassis until the latches click into place. Secure the card using the Phillips screws.

5 Replace the RF cables and power connections.

Replacing the Up-Converter

Use this procedure to replace the up-converter, if necessary.

> *Note:* It is possible to use the C3 without an up-converter card, by using the EXT UPCONV connector on the CPU and an external up-converter. The RF output at the EXT UPCONV jack has an output frequency of 44 MHz for North American DOCSIS, and 36.125 MHz for EuroDOCSIS.

Requirements

Contact your ARRIS representative to obtain a new up-converter.

Replacing the up-converter requires a complete shutdown of the C3 CMTS.

Use anti-static precautions such as a wrist grounding strap grounded to a grounded work area when handling the up-converter card.

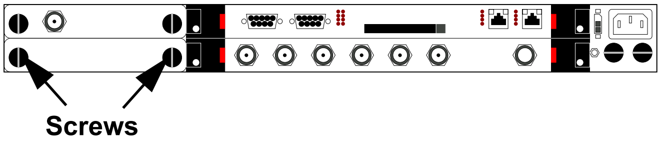
Action

1 Power down the CMTS by removing all power leads from the rear sockets.

2 Disconnect the downstream RF cable from the up-converter.

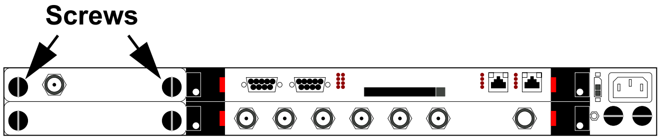
DANGER
Risk of equipment damage
If you do not remove the bottom slot cover before removing the up-converter, you risk breaking off surface-mount components on the bottom of the up-converter board during removal or installation.

3 Remove the bottom slot cover by loosening the two captive screws securing the slot cover to the chassis. Set the cover aside.



4 Remove the upstream card from the CMTS chassis as follows:

a Loosen the two captive screws securing the up-converter to the chassis.



b Grasp the up-converter by the provided handle and slide the card out of the chassis.

5 Install the new up-converter into the chassis. Line up the up-converter with the guides inside the chassis, and slide the card into the chassis. Secure it with the captive screws.

6 Replace the bottom slot cover.

7 Replace the RF cable and power connections.

Replacing Fuses

Use this procedure to replace the fuses. The C3 CMTS has two fuses, located beneath the power connectors on the back of the CMTS chassis.

Requirements

Replace F1 (AC fuse) only with: 250V/5A Antisurge (T) Glass.

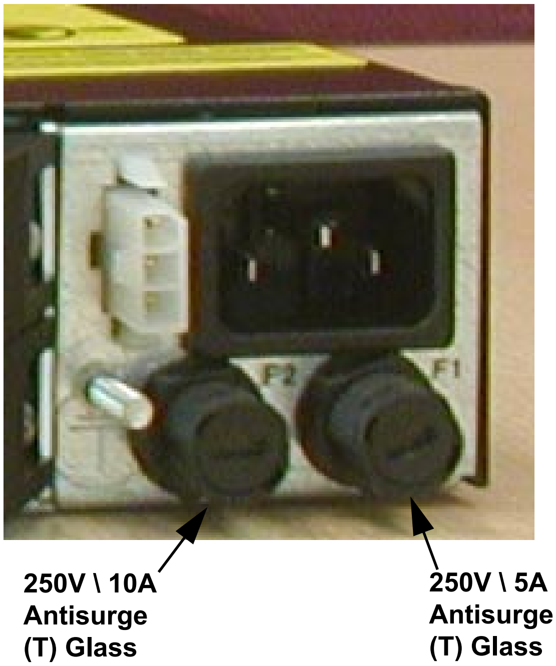
Replace F2 (DC fuse) only with: 250V/10A Antisurge (T) Glass.

CAUTION
Risk of fire

For continued protection against risk of fire, replace only with same type and ratings of fuses.

Diagram

The following diagram shows the fuse locations.



Resetting the CMTS after Thermal Overload

If a thermal overload occurs, the C3 shuts down safely with no damage. The power supplies are disabled and remain in an interlocked state until you clear the interlock manually.

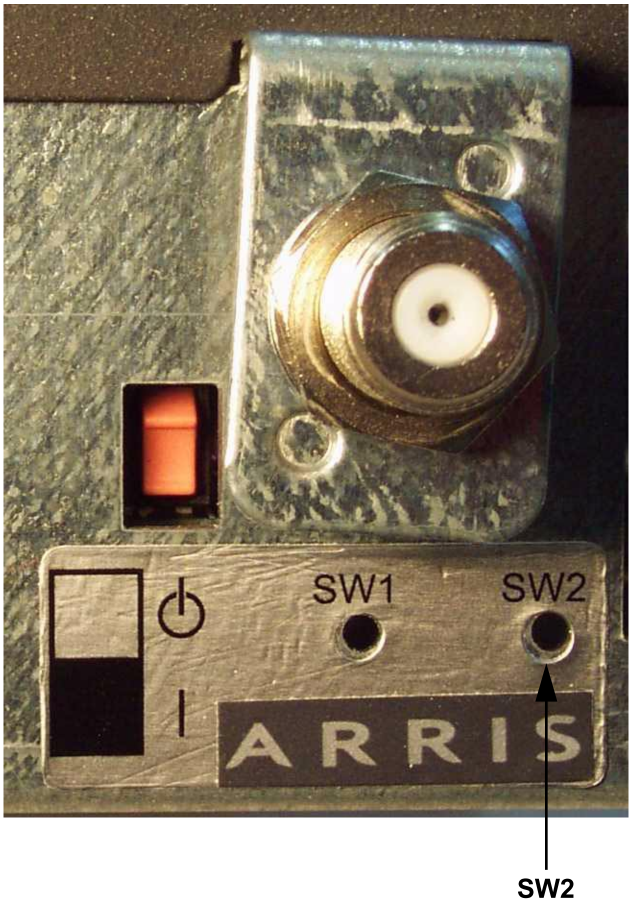
Action

Follow these steps to clear the interlocked state.

1 Correct the condition that caused the thermal overload.

2 Remove the C3 front panel as described in “Front Panel Removal and Replacement” on page 9-2.

3 Locate the switch SW2, under the RF test jack on the right side of the C3. The following photo shows its location.



Note: SW1 is the reset for the environmental monitoring CPU and should never be needed.

4 Press SW2 to clear the thermal overload interlock condition.

Upgrading the CMTS Software

The C3 can boot from a software image located on its local Compact Flash disk, or from an image on a TFTP server. Use this procedure to upgrade a C3 CMTS to the current software version and set the booting method.

Booting Methods

The C3 supports the following booting methods:

- Local boot—the C3 loads and runs a software image located on its Compact Flash disk.
- Network boot—the C3 loads and runs a software image located on a TFTP server.

Requirements

Before performing this procedure, you need the upgrade software image. Contact your ARRIS representative for information about obtaining the upgrade software image.

For network booting, you must have an operating TFTP server containing the software image file that the C3 downloads at boot time. For best results, the TFTP server in question should be located on the same LAN (and preferably on the same hub) as the C3. Close location minimizes the possibility that a network failure could prevent the C3 from booting properly.

CAUTION
Service affecting
Upgrading the C3 requires a reboot to load the new software image. To minimize disruption of service, perform the reboot only during a scheduled maintenance window.

During the upgrade process, avoid using the **write erase** command to erase the startup configuration. While the C3 would create a new default startup configuration, the default does not include CLI accounts and passwords. Therefore, telnet access is disabled and you would need to use the serial console to restore the CLI accounts.

Action

Perform the following tasks as needed.

Copying the Image Over the Network

Follow these steps to upgrade the C3. This procedure uses the IP address **10.1.12.5** and the file name **C3_v03.00.01.27** as examples; replace them with the IP address of your TFTP server and the actual software load file name.

1 Log into the C3 console and enter privileged mode, if you have not already done so.

```
Login: xxxxxxx

Password: xxxxxx

C3>enable

Password: xxxxxx

C3#
```

2 Enter the following commands to copy the new software image onto the C3:

```
C3#copy tftp flash

Address or Name of remote host []? 10.1.12.5

Source filename []? C3_v02.00.03.08.bin

Destination filename [C:/C3_v03.00.01.27.bin]? <enter>
Accessing tftp://10.1.12.5/C3_v03.00.01.27.bin...
Load C3_v03.00.01.27.bin from tftp://10.1.12.5
:!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!
[OK - 8300967 bytes]

8300967 bytes copied in 25 secs (332038 bytes/sec)

C3#dir

Listing Directory C:/:
-rwxrwxrwx  1 0      0            690 Jul 15 19:56 autopsy.txt
-rwxrwxrwx  1 0      0            996 Jun 19 14:40 root.der
-rwxrwxrwx  1 0      0          10901 Jul 15 19:56 snmpd.cnf
-rwxrwxrwx  1 0      0             45 Jul 16 16:35 tzinfo.txt
-rwxrwxrwx  1 0      0          19213 Jun 19 14:40 fp_uload.hex
```

```
-rwxrwxrwx  1 0       0          10764 Jul 15 19:55 startup-configuration
-rwxrwxrwx  1 0       0           5208 Jun 19 14:40 dfu_uload.hex
drwxrwxrwx  1 0       0           2048 Jun 26 18:31 CONFIG/
drwxrwxrwx  1 0       0           2048 Jul 15 16:38 SOFTWARE/
-rwxrwxrwx  1 0       0          10901 Jul 15 19:56 snmpd.cnf~
drwxrwxrwx  1 0       0           2048 Jun 19 15:07 Syslog/
-rwxrwxrwx  1 0       0        8001301 Jun 17 19:57 vxWorks.bin.img
-rwxrwxrwx  1 0       0          10764 Jul 15 19:55 startup-temp
-rwxrwxrwx  1 0       0         161251 Jul 15 19:55 shutdownDebug.log
-rwxrwxrwx  1 0       0           1258 Jul 23 16:08 tmp_file-0001
-rwxrwxrwx  1 0       0        8300967 Jul 23 16:08 C3_v02.00.03.08.bin
```

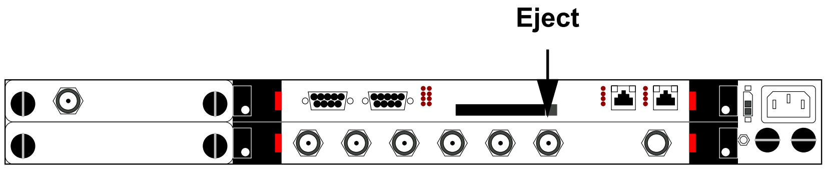
3 Proceed to “Configuring the C3 to Boot from the Flash Disk” on page 9-17.

Using a Compact Flash Reader

Instead of copying the software image over the network, you can eject the Compact Flash disk from the C3 and copy the image directly from another computer. You need a Compact Flash reader (and driver software, if not already installed) to perform this task. Follow these steps:

1 Attach the Compact Flash reader to your computer, if necessary.

2 Push the eject button to the right of the Compact Flash card on the back of the C3. The following figure shows the location of the eject button.



The console displays the message “interrupt: Compact Flash card removed”

Note: Removing the Compact Flash card from the C3 has no effect on normal operation. However, the C3 refuses all commands that would change the configuration or operation of the CMTS, or access the disk, until you replace the Compact Flash card.

3 Insert the Compact Flash card into your computer’s reader.

The result depends on your computer. MacOS X and Windows systems automatically mount the disk; most Linux or BSD systems require you to use the **mount** command as root to mount the disk.

4 Copy the new software image onto the Compact Flash disk.

5 Eject the Compact Flash card from your computer and insert it in the slot in the C3 rear panel.

The C3 console displays the messages "interrupt: Compact Flash Card inserted" and "C:/ - Volume is OK"

6 Proceed to "Configuring the C3 to Boot from the Flash Disk" on page 9-17.

Configuring the C3 to Boot from the Flash Disk

Follow these steps to configure the C3 for local booting. This procedure uses the file name **C3_v02.00.03.08** as an example; replace it with the actual software load file name.

1 Use the following commands to configure the C3 to boot from the image on the Compact Flash disk:

```
C3# configure terminal ↵
C3(config)# boot system flash C3_v02.00.03.08.bin ↵
C3# exit ↵
```

CAUTION
Service affecting
Perform the following step only during a scheduled maintenance window to minimize service disruptions.

2 During the maintenance window, reboot the C3 using the **reload** command:

```
C3#reload
Save configuration when rebooting(Y/N)?Y
Are you sure you want to reboot the CMTS(Y/N)?Y
Reload in progress.
CadantC3 shutting down
...
```

3 After the C3 finishes rebooting, log in and use the **show version** command to verify that it is running the correct software image:

```
C3>show version

ARRIS CLI version .02
Application image: 3.0.1.27, Jun 20 2003, 15:26:37
BootRom version 1.26
VxWorks5.4.2

...
```

The “Application image” shows the software image version currently running. If this does not correspond to the image on the compact flash disk, a configuration problem may be preventing the C3 from accessing the new load, or the load file itself may be corrupt.

Configuring the C3 to Boot from a TFTP Server

Follow these steps to configure the C3 for network booting. This procedure uses the IP address **10.1.12.5** and the file name **C3_v03.00.01.27** as examples; replace them with the IP address of your TFTP server and the actual software load file name.

1 Use the following commands to configure the C3 to boot from the image on the TFTP server:

```
C3# configure terminal ↵
C3(config)# boot system tftp C3_v03.00.01.27.bin 10.1.2.3 ↵
C3# exit ↵
```

CAUTION
Service affecting
Perform the following step only during a scheduled maintenance window to minimize service disruptions.

2 During the maintenance window, reboot the C3 using the **reload** command:

```
C3#reload
Save configuration when rebooting(Y/N)?Y
Are you sure you want to reboot the CMTS(Y/N)?Y
Reload in progress.
CadantC3 shutting down
.
.
.
```

3 After the C3 finishes rebooting, log in and use the **show version** command to verify that it is running the correct software image:

```
C3>show version

ARRIS CLI version .02
Application image: 2.0.3.8, Jun 20 2003, 15:26:37
BootRom version 1.26
VxWorks5.4.2
```

. . .

The “Application image” shows the software image version currently running. If this does not correspond to the image on the TFTP server, a network or configuration problem may be preventing the C3 from accessing the TFTP server at boot time.

Enabling Licensing Features

The C3 contains certain features that require a license key in order to be enabled and used. These features are RIP and Bridge Groups.

Requirements

Contact your ARRIS representative to obtain a key(s) for the feature(s) being implemented.

The host ID of the CMTS and the feature(s) to be implemented must be provided to ARRIS. The host ID can be obtained using the privileged command **hostid** or **show license**. If privileged mode is not available the **show version** command can be used. The ARRIS representative will then provide a key for each CMTS and each feature enabled within the CMTS.

Action

1 Obtain key from ARRIS representative.

2 Log into the CMTS and enter privileged mode.

3 Enter the key information for the feature being enabled using the **license key** command. Refer to Chapter 6 for command syntax.

4 To verify that the key has been accepted, the **show license** command can be used. An example of the output is:

```
C3#show license
----------------------------------------------------------------------
C3 - hostid 312 - Licensed Features

         * RIP              ARSVS01163
         * BRIDGE_GROUPS    ARSVS01164
----------------------------------------------------------------------
C3#
```

5 If the feature needs to be disabled for any reason the **license remove** command may be used. Refer to Chapter 6 for command syntax.

Upgrading Dual Upstream Receivers

This procedure outlines the steps necessary to add a second or third dual upstream receiver to a MAC/PHY card. It is assumed in this procedure that one dual receiver card is already installed. Dual receiver cards should be populated from left to right.

Requirements

Prior to starting the upgrade procedure, ensure that you have the following:

- the upgrade hardware ordered from ARRIS
- torque driver with a size 0 Phillips head bit capable of measuring .2 Nm (28 oz-in).
- torque driver with a size 1 Phillips head bit capable of measuring .2 Nm (28 oz-in).
- 3/8-=32X3/32 1/2 Hex nut head for torque driver
- thread locking compound

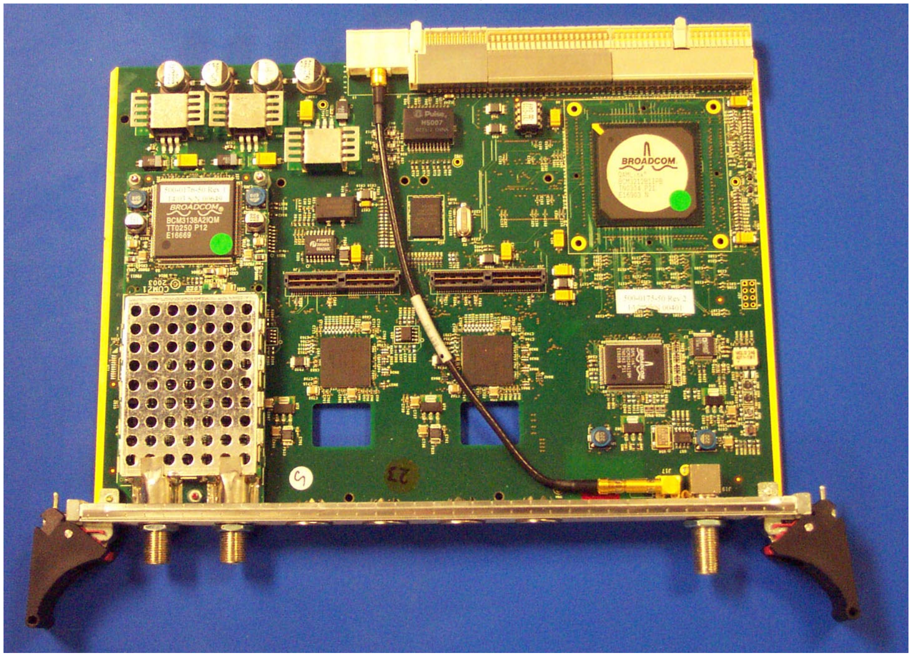
The following torque setting should be followed:

- required torque for nut 3/8 - 32 x 3/32 1/2 hex is 1.75 Nm (15.5 lb-in)
- required torque for nut M2 std thin steel zinc is .2 Nm (28 oz-in)
- required torque for nut M2.5 std thin steel zinc is .2 Nm (28 oz-in)

Action

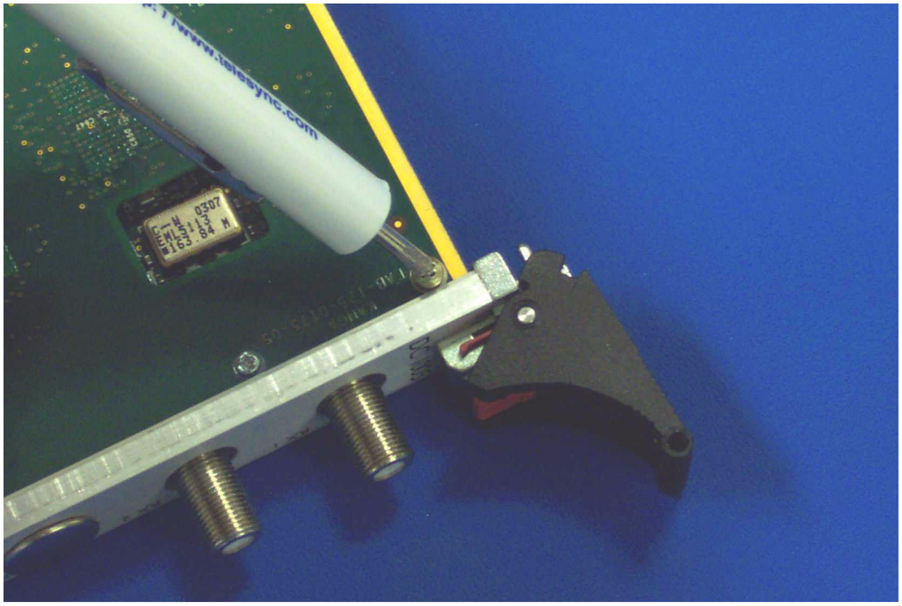
1. Remove the MAC/PHY as outlined in procedure“Replacing the RF Card” on page 9-8.

2 The procedure will begin with a MAC/PHY board populated as below. Remove any blanking plugs from the face plate.

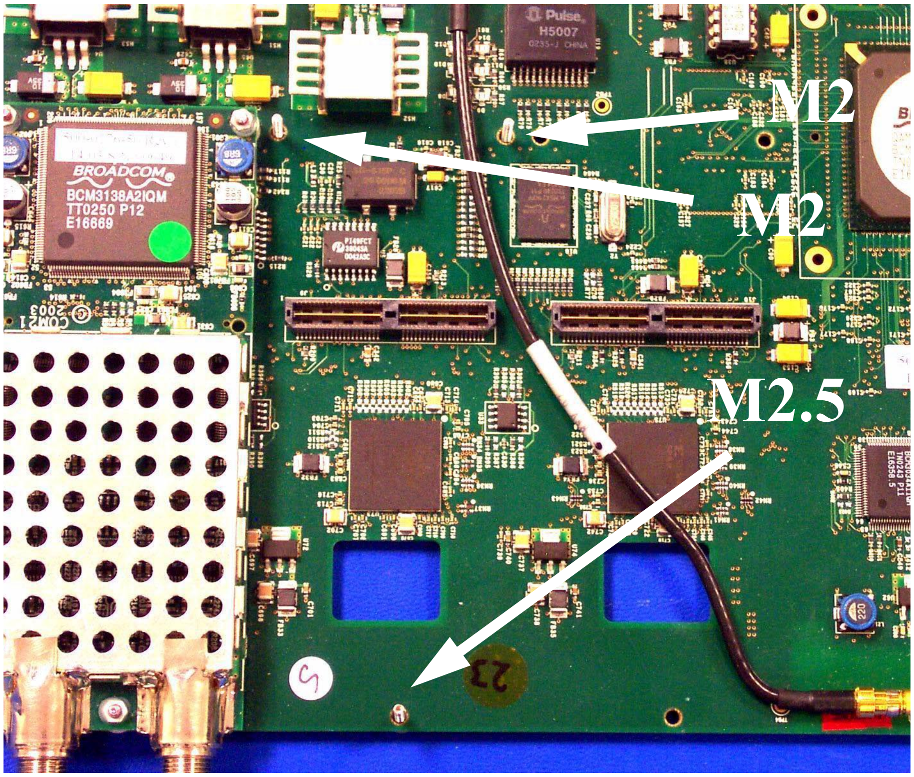


3 Remove all nuts and washers from the front panel.

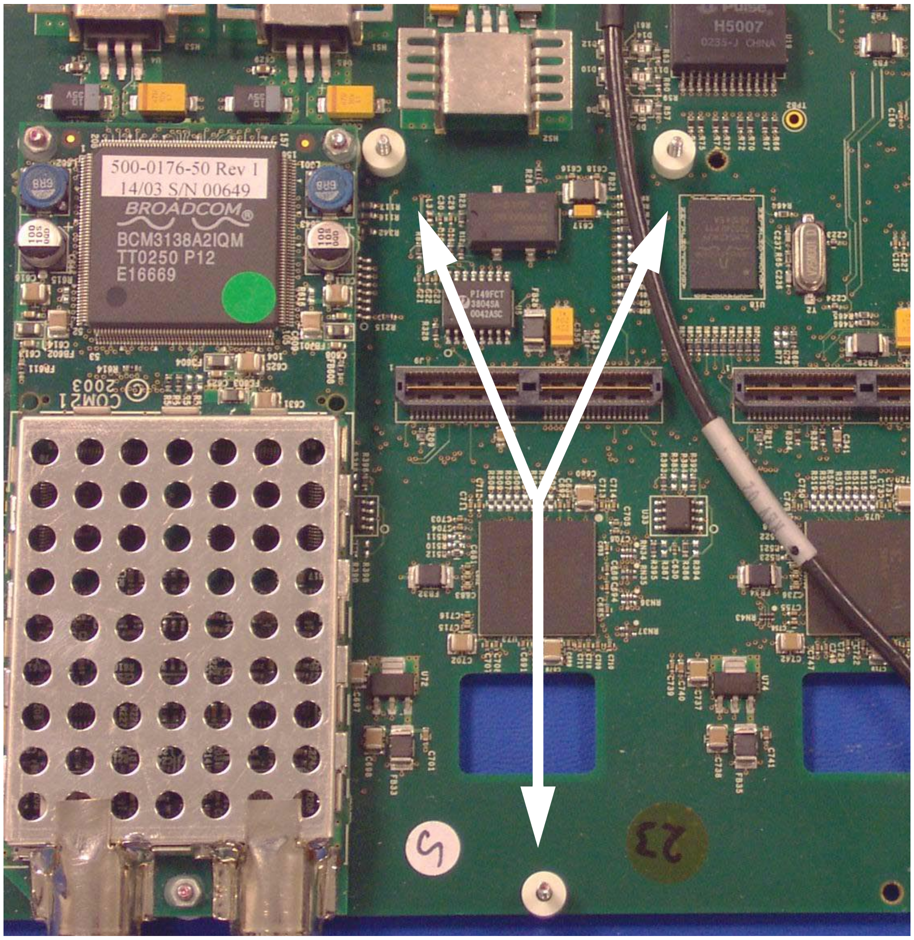
4 Turn the board over and remove the two screws and washers securing the faceplate to the printed circuit board (PCB) and remove the faceplate. If there is an insulation sheet on the underside of the board, bend it back carefully (do not fold).



5 Take the three screws and thread them through the underside of the MAC/PHY card. Be sure to place the M2.5 screw only in the position noted in the figure below.

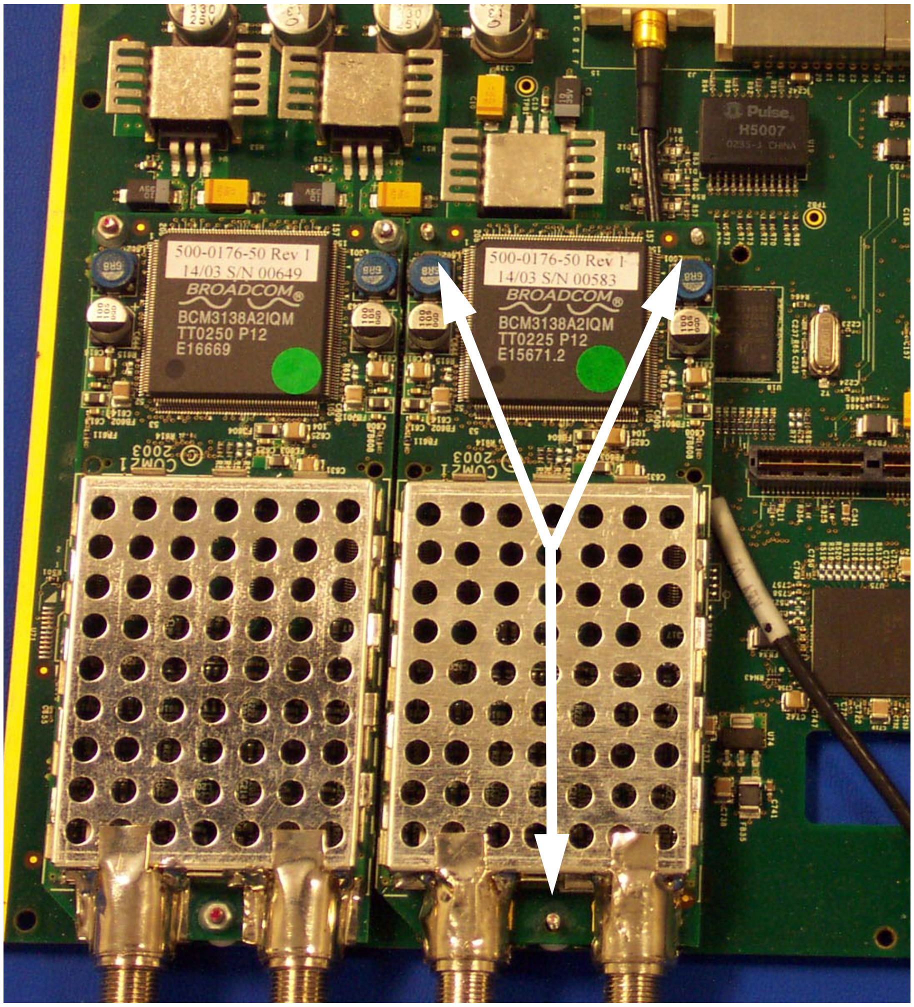


6 With the three screws showing, place the nylon stand offs on the three screws as shown below.



7 Place the dual upstream module into position with the three screws protruding from the associated holes on the MAC/PHY card. The dual upstream module should be installed such that the nylon stand offs fill the gap between the two boards exactly. The image below

shows the dual upstream module positioned correctly. Note the nuts have not been placed on the screws yet.



8 Take an M2.5 screw and nylon washer and place the washer over the protruding screw head. This screw is **only** to be used on the hole which is closest to the front of the board.

9 Place a dab of thread locking compound on the top of the screw. Put the M2.5 nut on the screw and hand tighten a torque value of .2 Nm (28 oz-in). using the size 1 Phillips screwdriver.

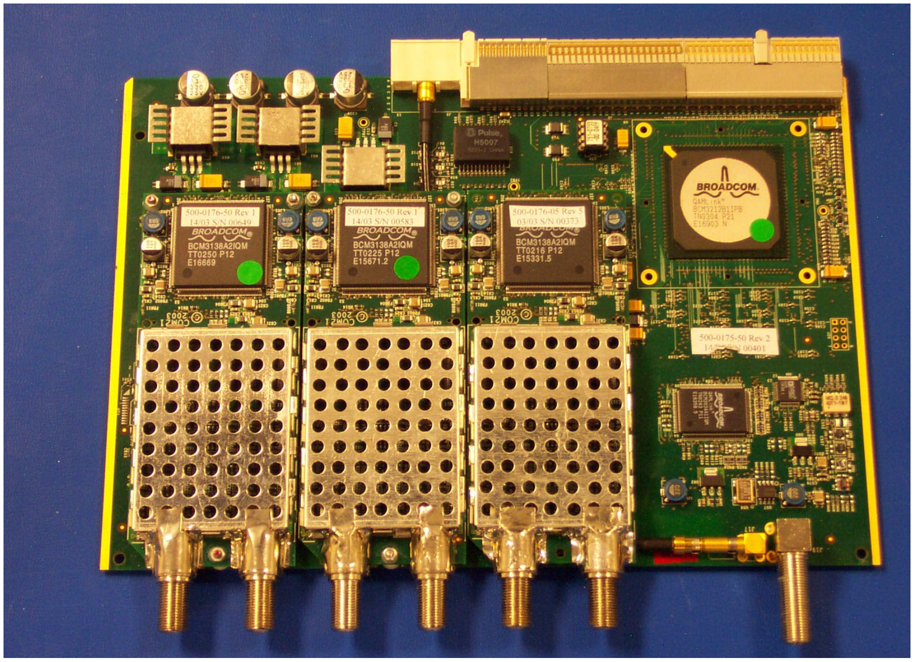
10 Steps 8 and 9 should be repeated using the **M2** screws and nuts for the other two standoffs/points on the dual upstream module. Tighten using the size 0 screwdriver to a torque value of .2 Nm (28 oz-in). The dual upstream module should now be secure as shown in the figure below. Take note of where the M2 and M2.5 screws and washers are positioned as shown in step 4.

11 At this point in the procedure, another dual upstream module may be added or the face plate replaced.

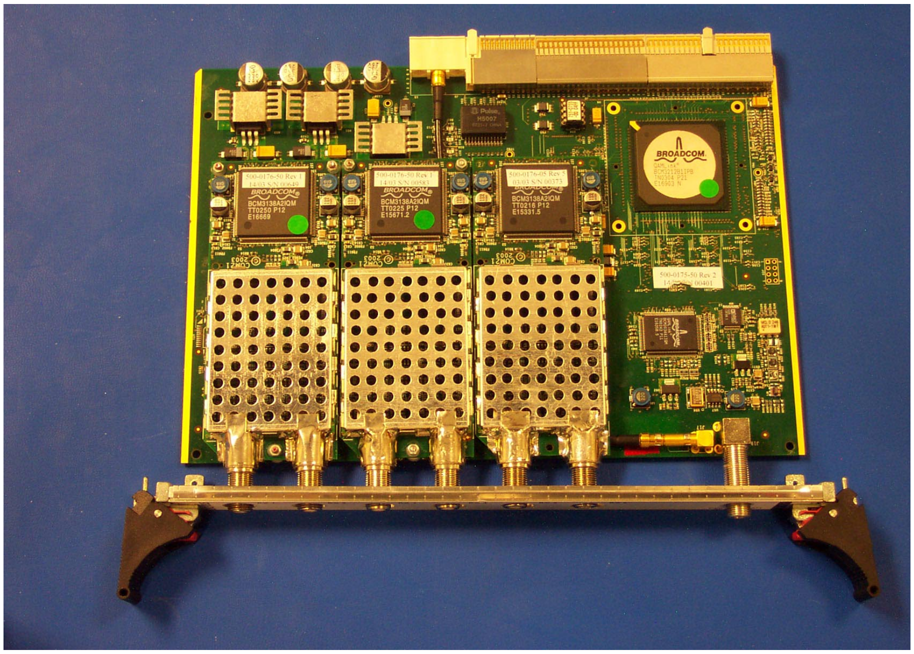
Note: If another dual upstream module is being added, care should be taken to ensure that the IF cable is routed as shown in the figure in step 5 above. Notice how the cable is pushed close to the edge of the PCB cutout.

It is possible to pinch the cable between the board edge and components on the base of the third dual upstream module. For this reason, care should be taken when adding a third module.

12 The addition of a third dual upstream module is identical to that of the second, having taken the IF cable routing into consideration.

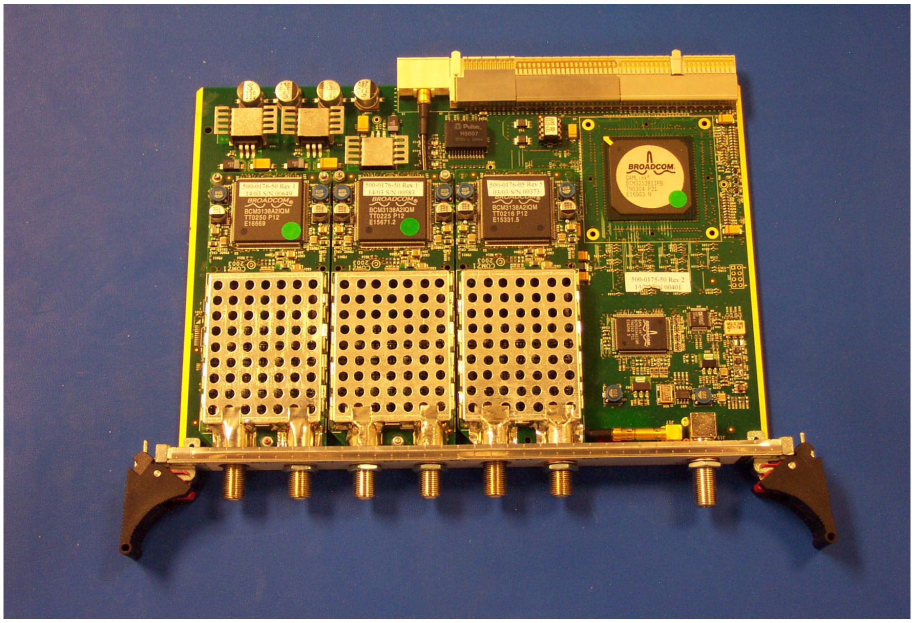


13 To assemble the face plate, the procedure is the opposite of disassembly. Place the face plate over all F-connectors and slide into place as shown in the figure below.



14 Secure the face plate to the PCB using the screws and washers removed in the earlier step and tighten to a torque of .6 Nm (5.2 lb-in). If there is an insulation sheet on the underside of the board, tuck it under the face-plate.

15 Secure all F connectors to the face plate using a lock washer and a hex nut, tighten to 1.75 Nm (15.5 lb-in). The receiver should now be completed as in the figure below. If only 2 dual upstream modules are present, fill the unpopulated upstream holes with blanking plugs.



16 Replace the RF card into the C3 using the procedure “Replacing the RF Card” on page 9-8.

A Specifications

This appendix lists specifications for the ARRIS Cadant C3 CMTS.

Product Specifications

8,000 Unicast service identifiers (SIDs)

Dual 10/100/1000BT Network Interfaces

Management interface: command-line interface for system configuration and management tools (telnet, SNMP)

Physical Interfaces

10/100/1000-Base T—Data

10/100/1000-Base T—Out-of-band management

1 downstream, 2 to 6 upstream RF (F-connector)

Serial console port

F-connector (test) on front panel

Logical Interfaces

Sub-interfaces:

Sub-interfaces	Capacity	
	Default	Advanced Bridging
Per physical interface	64	64
Entire CMTS	3	192
Per bridge group	3	10

Private cable VPNs: up to 64 (one per cable sub-interface) with CPE membership specified by CMTS configuration or by modem provisioning system

IP addresses per sub-interface: up to 16 (primary + 15 secondary)

Bridge groups (default operation): up to 2

Bridge groups (Advanced Bridging): up to 64

Protocol Support

Layer 2 bridging with static routing (up to 128 static routes) and DHCP relay

Layer 3 IP routing with RIPv1 and RIPv2

Hybrid Level 2/Level 3 operation

802.1Q VLAN support on cable and fastethernet sub-interfaces; each sub-interface can have:

- one configured VLAN specification
- up to 4 additional tags specified in a bridge bind

DHCP relay in layer 2 (bridging) and layer 3 (IP routing) mode:

- up to 3 types of DHCP helper address per sub-interface and up to 5 addresses per type
- support for DHCP relay address update based on cable modem or host DHCP request
- support for DOCSIS option 82 update

IGMPv2 proxy

Regulatory and Compliance

EMC: FCC Part 15 Class A, CE

DOCSIS: 1.1 qualified

Electrical Specifications

AC Power: 115 to 240 VAC, 2A, 47-63 Hz

DC Power: –40 to –60 V, 4A

Power consumption: 80 watts maximum

Redundant powering available—the C3 requires only one power supply to operate, but can be configured with two power supplies (DC and/ or AC) for load sharing and automatic fault recovery

Fuse F1: (AC fuse): 250V/5A Anti-surge (T) Glass

Fuse F2: (DC fuse): 250V/10A Anti-surge (T) Glass

Physical Specifications

19 in (W) x 18.3 in (D) x 1.75 in (H)

48.3 cm (W) x 46.5 cm (D) x 4.4 cm (H)

Height: 1 RU (rack unit)

Weight: 10 Kg

Environmental Specifications

Operating Temperature: 0° to 40° C

Storage Temperature: –40° to +75° C

Humidity: 10% to 80% non-condensing

Electromagnetic: FCC Part 15 Class A, CE

MTBF (excluding fans): 40,000 hours at 25°C based on accelerated life testing

RF Specifications

Upstream

Number of Upstreams: 2, 4, or 6

Frequency Range: 5 to 42 MHz (DOCSIS);
5 to 65 MHz (EuroDOCSIS, Japan DOCSIS)

Modulation: QPSK, 8QAM,16QAM, 32QAM, 64QAM, 128QAM and 256QAM.

Symbol Rate: 160, 320, 640, 1280, 2560, 5120 Ksymbol/sec

Data Rate: 5.12 to 40.96 Mbps (max)

Channel Bandwidth: 200, 400, 800, 1600, 3200, 6400 KHz

Receive Signal Level: –20 dBmv to +26 dBmV (valid level varies by symbol rate)

Downstream

Frequency range: 88 to 860 MHz

Modulation: 64 / 256 QAM

Data rate: 30 to 53.6 Mbps (max)

Transmit level: +45 to +61 dBmV

Output Impedance: 75 ohm

Modulation rate:

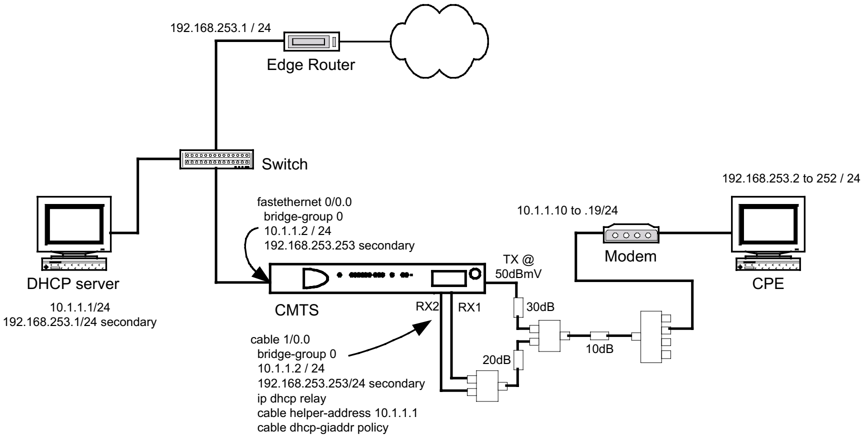
- 64 QAM: 5.056951 Msymbols/sec
- 256 QAM: 5.360537 Msymbols/sec
- EuroDOCSIS: 6.952Msymbols/sec

B CMTS Configuration Examples

This appendix provides the bare necessities to get an ARRIS Cadant C3 up and running with modems, and computers attached to modems, and a working DHCP server. It concentrates on the absolute minimal steps required to get a DOCSIS modem up and running after installing the C3.

Refer to Chapters 3 through 8 while following the examples in this appendix.

The most simple configuration is a cable modem, C3, and DHCP/TFTP server:



```
fastethernet 0/0.0
  bridge-group 0
  10.1.1.2 / 24
  192.168.253.253 secondary
```

```
cable 1/0.0
  bridge-group 0
  10.1.1.2 / 24
  192.168.253.253/24 secondary
  ip dhcp relay
  cable helper-address 10.1.1.1
  cable dhcp-giaddr policy
```

Note: Modems, CPE, and the DHCP server are all in the same subnet, and management traffic co-exists with user traffic.

C3 Install

Use the information in “Getting Started” (Chapter 1) and use the following information that is correct for the above network.

Set the C3 boot options as follows:

> ***Note:*** The firmware filename you are using may be different from the file shown in this example.

```
>bootCfg

Options:
*[1] Boot from TFTP
 [2] Boot from Compact Flash
Select desired option : [2]
Application Image path : [C:/ 3.0.1.27.bin]
CMTS Ip Address : [10.1.1.2]
CMTS Subnet Mask : [255.255.255.0]
TFTP Server Ip Address : [10.1.1.1]
Gateway Ip Address : [10.1.1.1]
Saving in non-volatile storage

>>
```

Confirm the boot options:

```
CMTS>bootShow
*** Current Boot Parameters ***
Boot from            : Compact Flash
Boot file            : C:/2.0.4.4.bin
CMTS IP Address      : 10.1.1.2
CMTS subnet mask     : ffffff00
Gateway Address      : 10.1.1.1
CMTS Name            : CMTS
Network port         : FE 0
Vlan Tagging         : Disabled
Vlan Id              : 1 (0x1)
CMTS>
```

> ***Note:*** If the “Network port” shows “FE 1,” use the **wan** command at the prompt to change this. Use **bootShow** again to confirm this change.

Use the following script to configure the C3 (this script assumes a factory default configuration). If not in a factory default condition, the factory default configuration can be restored by erasing the stored configuration (file name is **startup-configuration**) using **write erase** from privilege mode. Then issue a **reload** command, responding first with **no** and then **yes** to reboot. The C3 detects no startup-configuration file and re-creates it.

If the C3 has been used elsewhere in the past, this step is *highly* recommended as it may be simpler than inspecting and changing the current configuration.

Script example:

Copy this script to the clipboard, log on at the serial console CLI, entering privilege mode and using the Hyperterm Edit/paste to console.

```
! make sure in privilege mode before running
! this script
conf t
! enable basic snmp
snmp-server community public ro
snmp-server community private rw
!
! create account so telnet will work
cli account arris password arris
cli account arris enable-password arris
!
no ip routing
bridge 0
!
inteface fastethernet 0/0.0
bridge-group 0
ip address 10.1.1.2 255.255.255.0
ip address 192.168.253.253 255.255.255.0 secondary
management-access
exit
!
interface cable 1/0.0
bridge-group 0
! give cable interface ip address so dhcp relay will work
! can be the same as the management ip address as running
! in bridging mode
ip address 10.1.1.2 255.255.255.0
ip address 192.168.253.253 255.255.255.0 secondary
! turn on the upstreams
no cable upstream 0 shutdown
no cable upstream 1 shutdown
! Turn on DHCP relay so DHCP will be unicast to
! the required DHCP server
ip dhcp relay
cable helper-address 10.1.1.1
cable dhcp-giaddr policy
! turn on the downstream
no shutdown
```

```
exit
!
! for convenience during testing
! remove telnet session timeout
line vty
timeout 0
exit
exit
! save the configuration
write
```

At this point, the two green LEDS for Rx1 and Rx2 on the front panel are lit and the RF ports (upstream and downstream) are active.

If a modem is connected, it finds the downstream, ranges on an upstream, but fails at the DHCP stage. This is expected at this early stage.

DHCP Server Configuration

The DHCP server receives DHCP Discovers and Requests with a relay address (giaddr option) of **10.1.1.2** for cable modems and **192.168.253.253** for CPEs (hosts).

Any basic DHCP server with two defined scopes containing these subnets can issue an IP address for the modems and to the CPE.

The DHCP options provided to the modem should include the following:

Option name	Number	Description
min-lease-time max-lease-time	58 59	Default minimum (T1/renewal) and maximum (T2/rebinding) lease times
broadcast-address	28	Broadcast address for subnet to which client is attached
time-offset <int>	2	Time offset in seconds from UTC, positive going east, negative going west.
filename <name>	-	Sets the “file” field, which is the name of a file for the client to request from the next server, i.e. a modem configuration file
next-server <ip>	-	Sets the “siaddr” field, which defines the name of the next server (i.e. TFTP) to be used in the configuration process
bootfile-name	67	Name of bootfile to use when “file” field is used to carry options
tftp-server-name	66	Name of TFTP server to use when “sname” field is used to carry options.

Option name	Number	Description
routers <ip>	3	Router address for modem
time-servers <ip>	4	Time servers (as specified in RFC868)
log-servers <ip>	7	MIT-LCS log servers

The options use may depend on the selected DHCP server.

One additional step is required in the route table of the DHCP server in this example. The DHCP server must be given a gateway for the 192.168.253.0 network so that the DHCP Offer and Acks can be sent back to the CPE relay address.

TFTP Server Configuration

For the modem to boot completely, an accessible TFTP server as specified by the “siaddr” DHCP option and the boot-file or filename specified in the DHCP options must be resident in the TFTP server root folder.

Debug—What to Do if DHCP Not Working

If the DHCP server is located past a router on the operator backbone make sure that the DHCP server workstation can be pinged from the Cadant C3 CLI and that the Cadant C3 management address (10.1.1.2 in the above example) can be pinged from the DHCP server.

If secondary subnets exist on the Cadant C3, makes sure that these IP addresses can be pinged from the DHCP server. Note that “management-access” will have to be specified on the relevant sub-interfaces.

If the DHCP does not reach the DHCP server you should check the Cadant C3 configuration and specifically check (in the above example):

```
cable helper-address 10.1.1.1
```

On the C3, use the **debug** command to watch DHCP events on the cable modem and attached CPE:

```
! get modem mac address x.x.x.x that might be having dhcp issues
! for CPE dhcp debug still use cable modem mac address
show cable modem
! now turn on debug for selected modem
debug cable mac-address x.x.x.x [ verbose ]
debug cable dhcp-relay
term mon
```

Watch the console for DHCP:

- discover
- offer
- request
- ack (on the C3)

Note: If CPE DHCP is to be monitored, enable DHCP debug for the attached cable modem MAC address NOT the CPE MAC address.

See also: Chapter 7, “Managing Cable Modems,” and the section on DHCP.

Common Configurations

The following configurations provide C3 configuration from a factory default condition and in the more complicated examples, DHCP server configuration details.

Simple Bridging

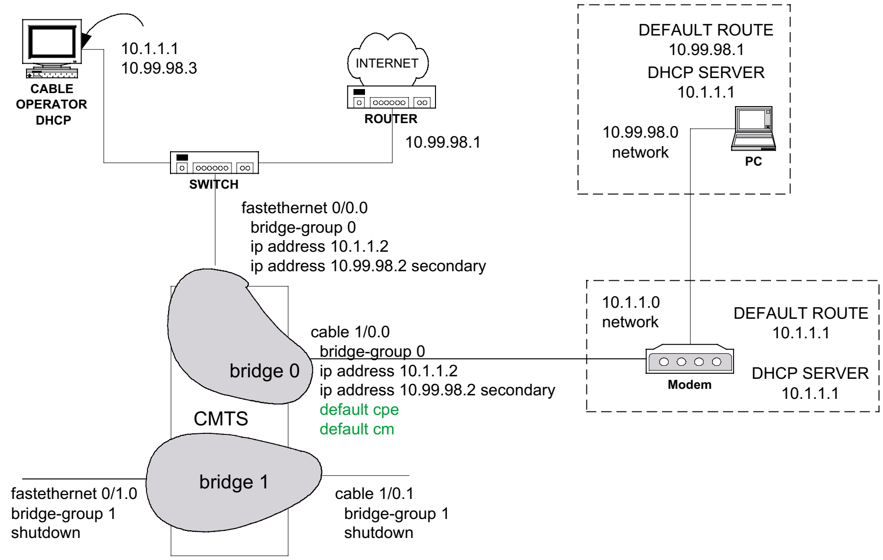
In a factory default configuration, the C3 is configured with two bridge groups, only one of which is active.

- fastethernet 0/0.0 and cable 1/0.0 are members of bridge group 0
- cable 1/0.1 is pre-defined
- cable 1/0.1 and fastethernet 0/1.0 are both members of bridge group 1
- cable 1/0.1 is shutdown
- default-cm-subinterface cable 1/0.0
- default-cpe-subinterface cable 1/0.0

All traffic uses the fastethernet 0/0 (WAN) interface.

This configuration is the equivalent of v2.0 series software “inband-management” operation.

The following examples repeat the simple example given above but showing in a more diagrammatic form the default allocation of sub-interfaces to the default bridge groups.



C3 Configuration

The following commands configure the C3 for simple bridging operation.

```
! make sure in privilege mode before running
! this script
conf t
! enable basic snmp
snmp-server community public ro
snmp-server community private rw
!
! create account so telnet will work
cli account arris password arris
cli account arris enable-password arris
!
no ip routing
! this bridge-group is already defined
bridge 0
!
inteface fastethernet 0/0.0
bridge-group 0
ip address 10.1.1.2 255.255.255.0
ip address 10.99.98.2 255.255.255.0 secondary
management-access
exit
!

interface cable 1/0.0
```

```
bridge-group 0
! give cable interface ip address so dhcp relay will work
! can be the same as the management ip address as running
! in bridging mode
ip address 10.1.1.2 255.255.255.0
ip address 10.99.98.2 255.255.255.0 secondary
! turn on the upstreams
no cable upstream 0 shutdown
no cable upstream 1 shutdown
! do not broadcast dhcp as we do not know
! what else is out there
ip dhcp relay
cable helper-address 10.1.1.1
cable dhcp-giaddr policy
! turn on the downstream
no shutdown
exit
!
! for convenience during testing
! remove telnet session timeout
line vty
timeout 0
exit
exit
! save the configuration
write
```

Simple Bridging with Separate Management Traffic

It is possible to configure the C3 using the factory default bridge groups and sub-interfaces to separate management traffic from other network traffic:

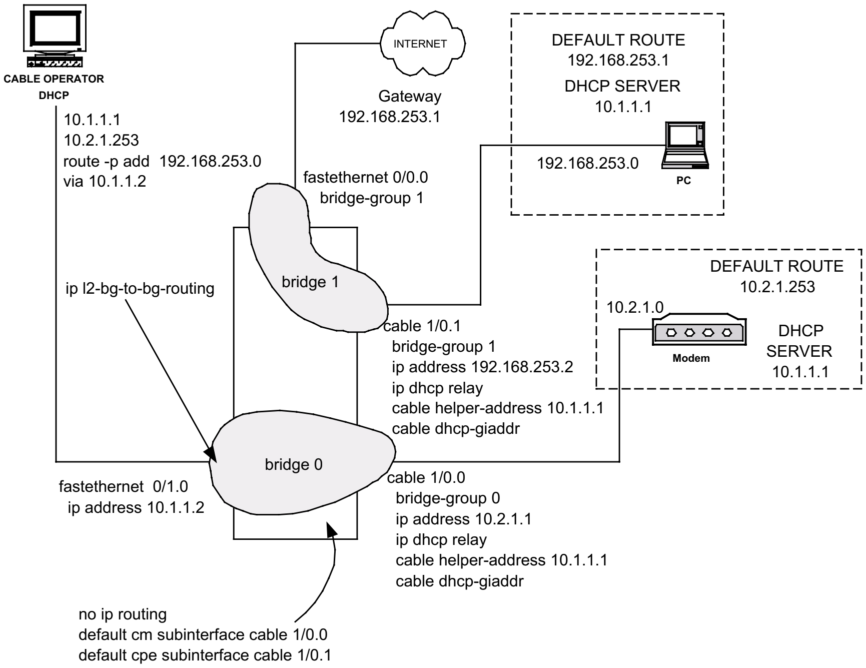
- fastethernet 0/1 and cable 1/0 are members of bridge group 0
- cable 1/0.1 is pre-defined
- cable 1/0.1 and fastethernet 0/0 are both members of bridge group 1
- default-cm-subinterface cable 1/0
- default-cpe-subinterface cable 1/0.1

Note: If the boot options network interface is changed to the fastethernet 0/1.0 sub-interface on first power up (no startup-configuration file exists) using the **mgmt** boot option command, this configuration is the resulting default.

The following example shows how the bridge group capability of the Cadant C3 can be used to completely isolate CPE traffic, including CPE broadcast traffic, from the management network.

The following example:

- makes use of the **default cm subinterface** and **default cpe subinterface** commands to map all CPE and modem traffic to separate cable sub-interfaces and hence to separate bridge groups and hence separate fastethernet sub-interfaces
- DHCP relay is being used for CPE and relies on the ability of the C3 to forward DHCP across bridge groups as long as **ip dhcp relay** is turned on in the bridge groups concerned.
- The specification **ip l2-bg-to-bg-routing** on fastethernet 0/1.0 is required for DHCP Renew Acks to be returned to the CPE across the bridge groups. No other sub-interface requires this specification.
- Does not require VLAN tagging of data on the CPE network attached to the WAN port.



C3 Configuration

```
configure terminal
! turn on simple snmp access
snmp-server community public ro
snmp-server community private rw
!
no ip routing
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.1
!
```

```
! bridges already defined as factory default
! bridge 0
! bridge 1
!
interface fastethernet 0/0.0
bridge-group 1
! no ip address
no shutdown
no management-access
exit
!
interface fastethernet 0/1.0
bridge-group 0
! define management ip address
ip address 10.1.1.2 255.255.255.0
! need to allow bg to bg routing so cpe DHCP
! renew ack can be forwarded back to bg 1
ip l2-bg-to-bg-routing
no shutdown
!
interface cable 1/0.0
bridge-group 0
ip address 10.2.1.1 255.255.255.0
! all modem traffic will default here
! IMPORTANT: DHCP server must have static route
! to this interface via the management interface
! to allow CM DHCP to be routed back here
ip dhcp relay
cable helper-address 10.1.1.1
cable dhcp-giaddr

interface cable 1/0.1
! all CPE traffic will default here
bridge-group 1
! must have some form of vlan tagging
! use "native" format
encapsulation dot1q 99 native
ip address 192.168.253.2 255.255.255.0
! IMPORTANT: DHCP server must have static route
! to this interface via the management interface
! to allow CPE DHCP to be routed back here
ip dhcp relay
cable helper-address 10.1.1.1
cable dhcp-giaddr
 exit
!
exit
exit
!
write
```

Bridging, Separate Management Traffic, CM and CPE DHCP Servers

The following figure shows the same example as used above but in this case, an ISP based DHCP server manages CPE IP addresses.

This example shows complete separation between CPE traffic and modem plus CMTS traffic.

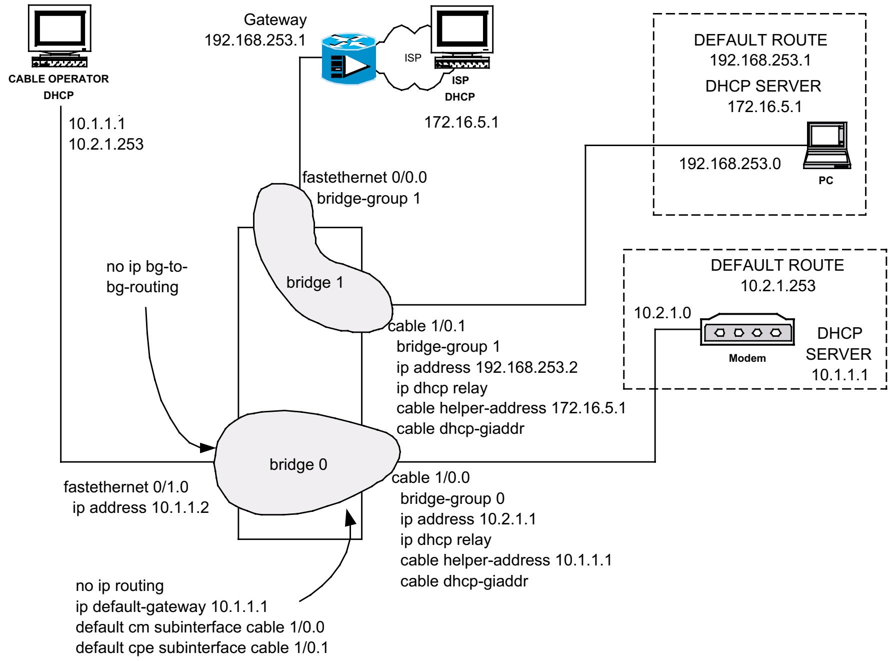
Variations from the previous example:

- now a separate **ip route** specification is used to tell the C3 how to find the ISP's 176.16.5.0 network.
- Fastethernet 0/1.0 no longer needs **ip bg-to-bg-routing**. The CPE DHCP Renew does not use this interface.

For example:

```
ip route 176.16.5.0 255.255.255.0 192.168.253.1
```

> ***Note:*** The fastethernet 0/0.0 sub-interface still does not need an IP address. Cable 1/0.1 has a 192.168.253.0 network address, so bridge group 1 is known to be attached to this IP network thus the C3 can find the specified route 192.168.253.1.



C3 Configuration

```
configure terminal
! turn on simple snmp access
snmp-server community public ro
snmp-server community private rw
!
no ip routing
```

```
ip route 172.16.5.0 255.255.255.0 192.168.253.1
default cm subinterface cable 1/0.0
default cpe subinterface cable 1/0.1
!
! bridges already defined as factory default
! bridge 0
! bridge 1
!
interface fastethernet 0/0.0
bridge-group 1
! no ip address
no shutdown
no management-access
exit
!
interface fastethernet 0/1.0
bridge-group 0
! define management ip address
ip address 10.1.1.2 255.255.255.0
! no need now as CPE dhcp never reaches this sub-interface
! but if dhcp server is not dual homed on cm subnet
! will still be needed for cm operation (as will static
! route in dhcp server to this interface for the modem
! network)
no ip l2-bg-to-bg-routing
no shutdown
!

interface cable 1/0.0
bridge-group 0
ip address 10.2.1.1 255.255.255.0
! all modem traffic will default here
ip dhcp relay
cable helper-address 10.1.1.1
cable dhcp-giaddr

interface cable 1/0.1
! all CPE traffic will default here
bridge-group 1
encapsulation dot1q 99 native
ip address 192.168.253.2 255.255.255.0
ip dhcp relay
cable helper-address 172.16.5.1
cable dhcp-giaddr
 exit
!
exit
exit
!
write
```

Advanced Bridging

An additional software licence is required to support the following examples. Please contact your account manager.

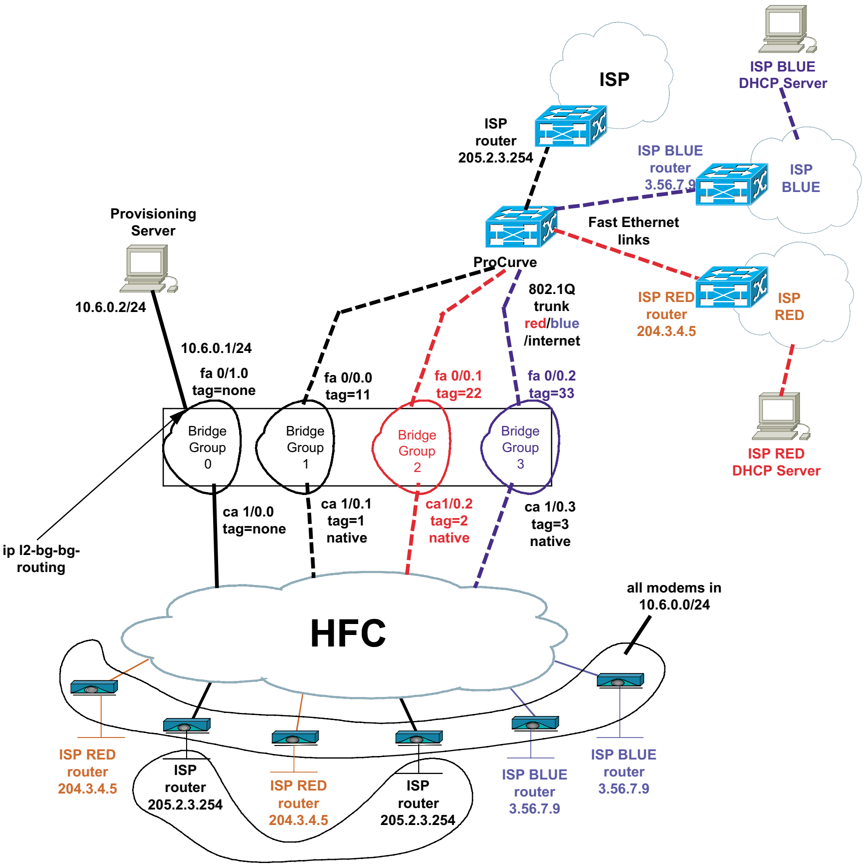
802.1Q VLAN Backbone

The advanced bridging and VLAN features of the Cadant C3 allow the use of more bridge groups, more sub-interfaces and more 802.1Q VLANs.

The following example shows an open access system implemented with a Cadant C3 in bridging mode with three ISPs. This example is shown as all the advanced bridging and VLAN abilities of the C3 are used.

The C3 can support up to 63 ISPs using this model.

In this example, two of the ISPs issue their own IP address; one ISP requires the cable operator to issue CPE IP addresses. In each case the router option passed to the CPE device is that of the ISP gateway router and is independent of the cable modem plant.



DHCP Server Configuration

To support this configuration the cable operator DHCP must have:

- A single scope defined for modems in the 10.6.0.0 network
- A scope defined for the network 205.2.3.0 network
- A method of providing specific DHCP options (including configuration file) for a specific modem (MAC address)

The modem DHCP Discover arrives at the DHCP server with its giaddr set to 10.6.0.1, so there must be an address pool for modems defined in the cable operator DHCP server for this subnet. For example, from 10.6.0.10 to 10.6.0.254.

Create a modem policy and assign to this address pool. This modem policy should have the DHCP server as the default route for the modems and should reference a suitable default set of DHCP options. This is the “default modem policy” for modems that have no other options specified (reserved).

The ISP’s DHCP Discover arrives at the operator DHCP server with a giaddr of 205.2.3.253.

> ***Note:*** You must enable **ip l2-bg-to-bg-routing** and management access on fastethernet 0/1.0 for CPE assigned to ISP to successfully renew the DHCP lease.

There should be a CPE address pool defined in the cable operator DHCP server for this subnet. For example, from 205.2.3.1 to 205.2.3.252.

The operator DHCP options in the policy for this address pool must have a router option of 205.2.3.254 (the internet gateway for ISP).

Important! The operator DHCP server needs a static route to the 205.2.3.0/24 network. Without this route, the DHCP server Offer and Ack responses to the CPE devices are not forwarded and DHCP Renew Ack to the CPE also fails. For example, **route -p add 205.2.3.0 mask 255.255.255.0 10.6.0.1**

The operator DHCP server needs to specify different configuration files for each modem depending on what the CPE attached to the modem is meant to be doing:

- Config file for “ISP” with VSE = 1
- Config file for “ISP RED” with VSE = 2
- Config file for “ISP BLUE” with VSE = 3

> ***Note:*** The default CPE sub-interface is specified as cable 1/0.1; thus any CPE traffic arriving via a modem with no VSE tagging

defaults to this sub-interface and ensuring that the CPE default allocation is to “ISP.”

The “ISP RED” CPE uses **ip dhcp relay** to reach the “ISP RED” DHCP server and “ISP BLUE” DHCP is broadcast through the C3 to the “ISP BLUE” DHCP server.

- Policy for internet ISP modems—configuration file referenced should have VSE=1
- Policy for internet ISP RED modems—configuration file referenced should have VSE=2
- Policy for internet ISP BLUE modems—configuration file referenced should have VSE=3

Reserve the modem MAC address in the appropriate address pool but OVERRIDE the default modem policy (defined above) with either:

- Policy for internet CPE modems—config file referenced should have VSE=1
- Policy for internet VPN RED—config file referenced should have VSE=2
- Policy for internet VPN BLUE—config file referenced should have VSE=3

This needs to be done per modem that is provisioned.

If a modem MAC address is not reserved in an address pool, it gets the default modem policy defined above using basic DHCP processing rules (matching giaddr to the available address pools). If the default for an un-provisioned modem is for Internet CPE, then this default policy should specify the configuration file that has a VSE=1.

DHCP for CPE devices attached to modems assigned to ISP RED or ISP BLUE are bridged and VLAN’d directly to the ISP backbones for processing.

C3 Configuration

```
! make sure in priv mode and in factory default
! before trying to paste the following
!
conf t
Bridge 0
Bridge 1
Bridge 2
Bridge 3
!
```

```
no ip routing
ip default-gateway 10.6.0.2
!
! ISP RED requires DHCP relay so tell the C3
! how to find the ISP RED dhcp server network
ip route 204.6.6.0 255.255.255.0 204.3.4.5
!
default cm sub interface cable 1/0.0
! set CPE default for ISP access
default cpe sub interface cable 1/0.1
!
interface fa 0/0.0
bridge-group 1
! no ip address required as bridging only
encapsulation dot1q 11
no management-access
exit
!
interface fa 0/0.1
bridge-group 2
! no ip address required as bridging only
encapsulation dot1q 22
no management-access
exit
!
interface fa 0/0.2
bridge-group 3
! no ip address required as bridging only
encapsulation dot1q 33
no management-access
exit
!
interface fa 0/1.0
bridge-group 0
! this is the C3 management IP address
ip address 10.6.0.1 255.255.255.0
management-access
! need this to allow CPE DHCP renew ack from DHCP server back to bg 1
ip l2-bg-to-bg-routing
exit

!
interface cable 1/0.0
! all modems are here by default
! enter RF config here
cable upstream 0 frequency 10000000
cable upstream 0 channel-width 3200000
cable upstream 0 ingress-cancellation
no cable upstream 0 shutdown
cable upstream 1 frequency 15000000
cable upstream 1 channel-width 3200000
cable upstream 1 ingress-cancellation
no cable upstream 1 shutdown
no shutdown
!
! Note can be the same as the management address
ip address 10.6.0.1 255.255.255.0
ip DHCP relay
cable helper-address 10.6.0.2 cable-modem
cable DHCP-giaddr primary
exit

!
interface cable 1/0.1
! for ISP CPE
bridge-group 1
! use this ip address to give giaddr to CPE DHCP discovers
! CPE should be given 205.2.3.254 as their gateway address
! and 205.2.3.254 should be the internet edge router
ip address 205.2.3.253 255.255.255.0
ip DHCP relay
cable helper-address 10.6.0.2
cable dhcp-giaddr primary
! VSE tag of 1 is required here
encapsulation dot1q 1 native
! turn on downstream privacy (BPI is on)
encapsulation dot1q 1 encrypted-multicast
! no cmts management allowed
no management-access
exit

!
interface cable 1/0.2
! for VPN RED
bridge-group 2
! need to use dhcp relay so set up
! ip addressing for relay to work
ip address 204.3.4.1 255.255.255.0
ip dhcp relay
cable helper-address 204.6.6.6
cable dhcp-giaddr primary
! VSE tag of 2 is required here
encapsulation dot1q 2 native
! give VPN members downstream privacy
encapsulation dot1q 2 encrypted-multicast
! allow l2 multicast and bcast echo
l2-broadcast-echo
l2-multicast-echo
! do not allow ip traffic to leave this bridge-group
no ip l2-bg-to-bg-routing
```

```
! no cmts management allowed
no management-access
! if required that VPN members get ip address from operator
! provisioning system
! add the following
! ip address 10.2.0.254 255.255.255.0
! ip DHCP relay
! cable helper-address 10.6.0.2
! cable DHCP-giaddr primary
exit
!
interface cable 1/0.3
! for VPN BLUE
bridge-group 3
! VSE tag of 3 is required here
encapsulation dot1q 3 native
! give VPN members downstream privacy
encapsulation dot1q 3 encrypted-multicast
! allow l2 multicast and bcast echo
l2-broadcast-echo
l2-multicast-echo
! do not allow ip traffic to leave this bridge-group
no ip l2-bg-to-bg-routing
! no cmts management allowed
no management-access
! if required that VPN members get ip address from operator
! provisioning system
! add the following
! ip address 10.3.0.254 255.255.255.0
! ip DHCP relay
! cable helper-address 10.6.0.2 host
! cable DHCP-giaddr primary
exit
```

Standard Ethernet Backbone

In the previous example, separate bridge groups are used for each ISP. This configuration however requires the use of an 802.1Q Ethernet backbone. In following example, 802.1Q VLANs are not used on the Ethernet backbone. This configuration is thus suitable for an operator that wishes to provide “open access” or “multi-ISP” without using 802.1Q backbone VLANs. The limitations of this configuration are:

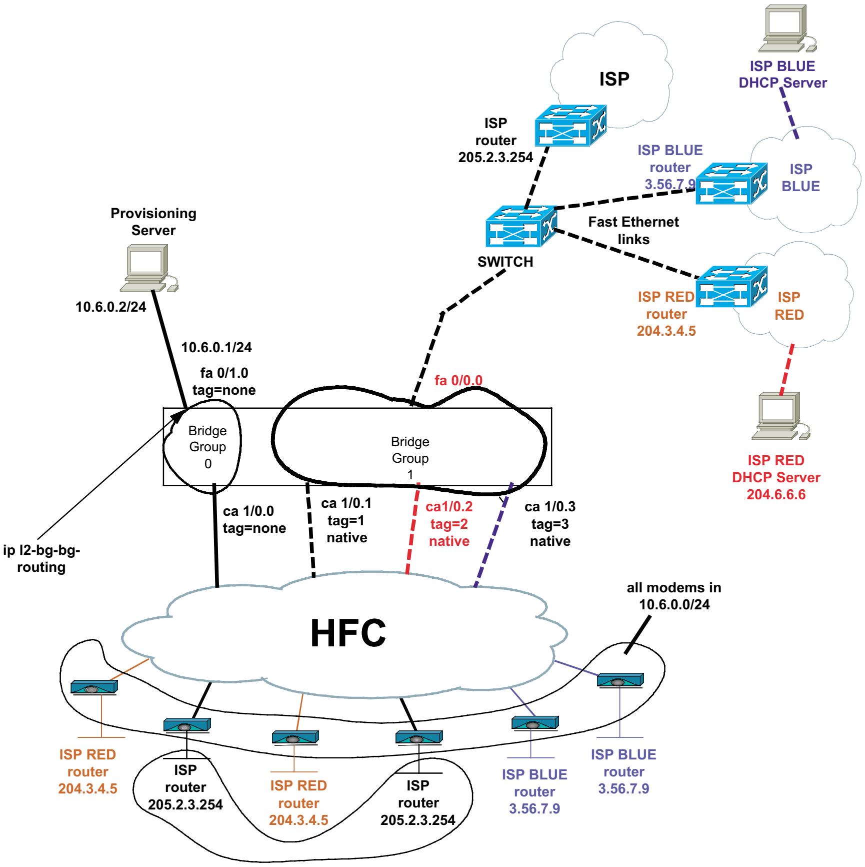
- the number of ISPs that can be supported in this manner is 9
- Since all CPE traffic shares the same bridge group, some protection is required to maintain separation between ISP traffic

The ability to add up to 10 sub-interfaces to one bridge group is being used, with this bridge group having one sub-interface connection to the operator Ethernet backbone.

All cable sub-interfaces are members of the same bridge group as fastethernet 0/0.

Other features to note in the following example:

- CPE traffic is still split into 3 native VLANs on 3 cable sub-interfaces using configuration file VSE allowing different specifications for each native VLAN e.g. ACL filters, DHCP relay etc.
- Downstream privacy is still turned on for each native VLAN.
- Again, one ISP uses the operator DHCP server for CPE DHCP; the other two ISPs use their own DHCP servers for CPE DHCP.
- Again, CPE should be given a default route of the respective ISP gateway router in the DHCP options.
- Up to 9 ISPs may be supported in this manner.



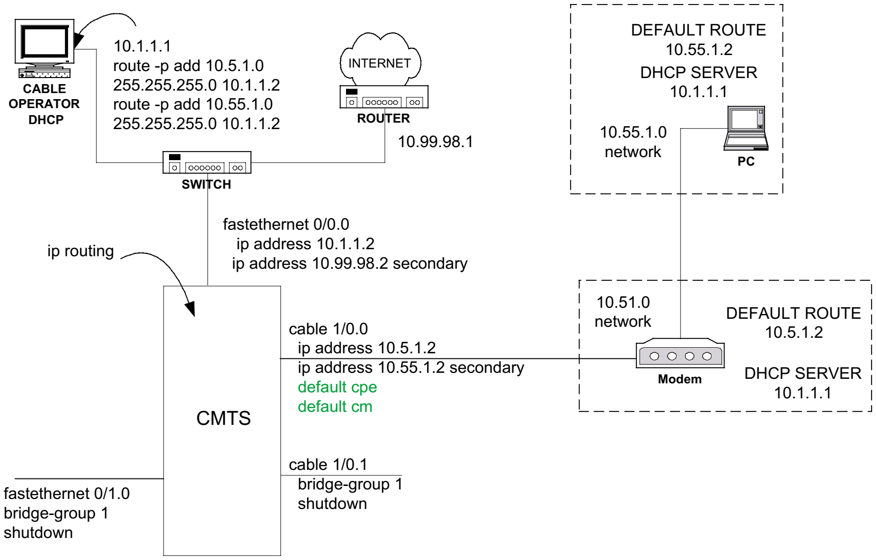
```
! make sure in priv mode and in factory default
! before trying to paste the   following
!
conf t
bridge 0
bridge 1
!
no ip routing
ip default-gateway 10.6.0.2
ip route 204.6.6.0 255.255.255.0 204.3.4.5
!
default cm sub interface cable 1/0.0
! set CPE default for internet access
default cpe sub interface cable 1/0.1
!
interface fa 0/0.0
bridge-group 1
! no ip address required as bridging only
no management-access
exit
!
interface fa 0/1.0
bridge-group 0
! this is the C3 management IP address
ip address 10.6.0.1 255.255.255.0
management-access
! need this to allow CPE DHCP RENEW ACK  from DHCP server back to bg 1
! and hence requesting CPE
ip l2-bg-to-bg-routing
exit
!

interface cable 1/0.0
bridge-group 0
! all modems are here by default
! enter RF config here
cable upstream 0 frequency 10000000
cable upstream 0 channel-width 3200000
cable upstream 0 ingress-cancellation
no cable upstream 0 shutdown
cable upstream 1 frequency 15000000
cable upstream 1 channel-width 3200000
cable upstream 1 ingress-cancellation
no cable upstream 1 shutdown
no shutdown
!
! Note can be the same as the management address
ip address 10.6.0.1 255.255.255.0
ip DHCP relay
cable helper-address 10.6.0.2 cable-modem
cable DHCP-giaddr primary
exit
!

interface cable 1/0.1
! for internet CPE
bridge-group 1
! use this ip address to give giaddr to CPE DHCP discovers
! CPE should be given 205.2.3.254 as their gateway address
! and 205.2.3.254 should be the internet edge router
ip address 205.2.3.253 255.255.255.0
ip DHCP relay
cable helper-address 10.6.0.2 host
cable dhcp-giaddr primary
! VSE tag of 1 is required here
encapsulation dot1q 1 native
encapsualtion dot1q 1 encrypted-multicast
! no cmts management allowed
no management-access
exit
!

interface cable 1/0.2
! for VPN RED
bridge-group 1
! need to use dhcp relay so set up
! ip addressing for relay to work
ip address 204.3.4.1 255.255.255.0
ip dhcp relay
cable helper-address 204.6.6.6
cable dhcp-giaddr primary
! VSE tag of 2 is required here
encapsulation dot1q 2 native
encapsulation dot1q 2 encrypted-multicast
! give VPN members downstream privacy
! allow l2 multicast and bcast echo
l2-broadcast-echo
l2-multicast-echo
! do not allow ip traffic to leave this bridge-group
no ip l2-bg-to-bg-routing
! no cmts management allowed
no management-access
! if required that VPN members get ip address from operator
! provisioning system
! add the following
! ip address 10.2.0.254 255.255.255.0
! ip DHCP relay
! cable helper-address 10.6.0.2 host
! cable DHCP-giaddr primary
exit
```

```
!

interface cable 1/0.3
! for VPN BLUE
bridge-group 1
! VSE tag of 3 is required here
encapsulation dot1q 3 native
! give VPN members downstream privacy
encapsulation dot1q 3 encrypted-multicast
! allow l2 multicast and bcast echo
l2-broadcast-echo
l2-multicast-echo
! do not allow ip traffic to leave this bridge-group
no ip l2-bg-to-bg-routing
! no cmts management allowed
no management-access
! if required that VPN members get ip address from operator
provisioning system
! add the following
! ip l2-bg-to-bg-routing
! ip DHCP relay
! cable helper-address 10.6.0.2 host
! cable DHCP-giaddr primary
exit
!
```

IP Routing

Simple Routing Network

This example is the equivalent of the bridging example given earlier in this chapter but in this case, bridge groups are not used—a pure routing model is used.



```
10.1.1.1
route -p add 10.5.1.0
255.255.255.0 10.1.1.2
route -p add 10.55.1.0
255.255.255.0 10.1.1.2
```

```
fastethernet 0/0.0
  ip address 10.1.1.2
  ip address 10.99.98.2 secondary
```

```
cable 1/0.0
  ip address 10.5.1.2
  ip address 10.55.1.2 secondary
  default cpe
  default cm
```

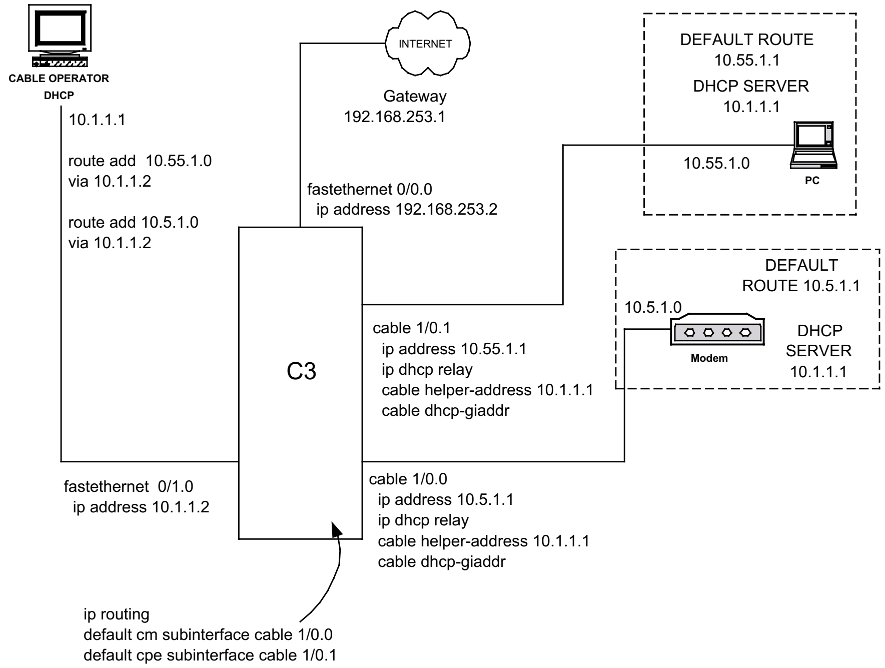
```
cable 1/0.1
  bridge-group 1
  shutdown
```

```
fastethernet 0/1.0
bridge-group 1
shutdown
```

```
! make sure in privilege mode before running
! this script
conf t
!
! provide default route for CPE
ip route 0.0.0.0 0.0.0.0 10.99.98.1
!
! enable basic snmp
snmp-server community public ro
snmp-server community private rw
!
! create account so telnet will work
cli account arris password arris
cli account arris enable-password arris
!
ip routing
!
inteface fastethernet 0/0.0
! remove the default bridge-group allocation
no bridge-group
ip address 10.1.1.2 255.255.255.0
ip address 10.99.98.2 255.255.255.0 secondary
management-access
exit
!
interface cable 1/0.0
no bridge-group
ip address 10.5.1.2 255.255.255.0
ip address 10.55.1.2 255.255.255.0 secondary
! turn on the upstreams
no cable upstream 0 shutdown
no cable upstream 1 shutdown
ip dhcp relay
cable helper-address 10.1.1.1
cable dhcp-giaddr policy
! turn on the downstream
no shutdown
exit
!
! for convenience during testing
! remove telnet session timeout
line vty
timeout 0
exit
exit
! save the configuration
write
```

Routing, Separate Management Traffic

Again, this example is the equivalent routing version of the simple bridging example presented above.



```
configure terminal
! turn on simple snmp access
snmp-server community public ro
snmp-server community private rw
!
! inband-managment
!
ip routing
!
! provide default route for CPE
ip route 0.0.0.0 0.0.0.0 192.168.253.1
!
default cpe subinterface cable 1/0.1
default cm subinterface cable 1/0
!
interface fastethernet 0/0.0
ip address 192.168.253.2 255.255.255.0
no bridge-group
no management-access
no shutdown
!
interface fastethernet 0/1
ip address 10.1.1.2 255.255.255.0
management-access
no shutdown
!
```

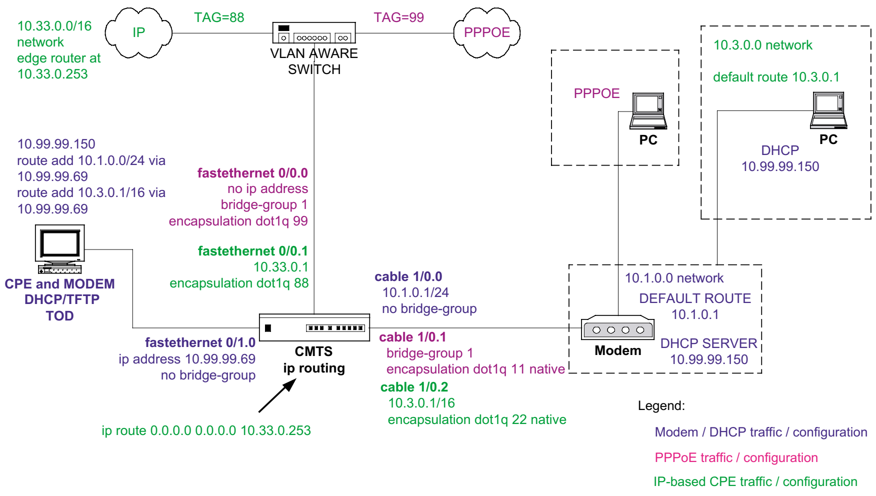
```
interface cable 1/0.0
no bridge-goup
ip address 10.5.1.1 255.255.255.0
ip dhcp relay
ip dhcp relay information option
cable dhcp-giaddr primary
cable helper-address 10.1.1.1
exit
!
interface cable 1/0.1
ip address 10.55.1.1 255.255.255.0
ip dhcp relay
ip dhcp relay information option
cable dhcp-giaddr primary
cable helper-address 10.1.1.1
no management-access
no shutdown
exit
!
exit
exit
```

Hybrid operation

The following example shows bridging being used to support CPE running at layer 2 (PPPoE) and IP routing being used to support CPE running at the IP level and Ethernet 802.1Q VLANS being used to separate traffic on the Ethernet backbone.

Note that bridging and routing is being performed by separate cable sub-interfaces. It is possible to both bridge and route using the one sub-interface.

Configuration file “VSE” is being used to map CPE traffic to sub-interfaces and hence to the capabilities of that sub-interface, either bridging or IP routing.



```
configure terminal
! turn on simple snmp access
snmp-server community public ro
snmp-server community private rw
!
cli account arris password arris
cli account arris enable-password arris
line vty
timeout 0
line console
timeout 0
exit
!
ip routing
! set default route for CPE ip traffic gateway
ip route 0.0.0.0 0.0.0.0 10.33.0.253
!
! factory defaults
! bridge 0
! bridge 1
!
interface fastethernet 0/0
bridge-group 1
! no IP address required
no shutdown
no management-access
encapsulation dot1q 99
exit
!
interface fastethernet 0/0.1
ip address 10.33.0.1 255.255.0.0
no shutdown
no management-access
encapsulation dot1q 88
exit
!
interface fastethernet 0/1.0
! management ip address of cmts
ip address 10.99.99.69 255.255.255.0
! make a routed sub-interface
no bridge-group
no shutdown
management-access
exit
!


interface cable 1/0.0
! for modems
! make a routed sub-interface
```

```
no bridge-group
no cable upstream 0 shutdown
no cable upstream 1 shutdown
no shutdown
ip address 10.1.0.1 255.255.0.0
no management-access
ip dhcp relay
ip dhcp relay information option
cable dhcp-giaddr primary
cable helper-address 10.99.99.150
exit
!
interface cable 1/0.1
! for PPPoE based CPE devices
! no ip address required
no management-access
bridge-group 1
encapsulation dot1q 11 native
exit
!
interface cable 1/0.2
! for IP based CPE devices
no bridge-group
ip address 10.13.0.1 255.255.0.0
encapsulation dot1q 22 native
no management-access
ip dhcp relay
cable helper-address 10.99.99.150
cable dhcp-giaddr primary
exit
!
exit
```

C

Factory Defaults

If no configuration is performed, the C3 uses the following default configuration.

Note that under default conditions, the downstream is turned off, no user accounts are defined (disabling telnet access until they are defined).

> ***Note:*** IP addresses shown following are network dependent and are set from the boot configuration.

Default Configuration Listing

```
C3#show config
!****Generated on WED FEB 25 10:37:13 2004

!****by S/W version 3.0.1.27
!
hostname "C3"
!
!boot system cur-flash

!
snmp-server contact "support@arrisi.com"
snmp-server location "3871 Lakefield Drive, Suite 300, Suwanee, GA 30024"
snmp-server engineboots 13
snmp-server view "default" "iso" included
snmp-server view "default" "snmpResearch" excluded
snmp-server view "default" "snmpTargetMIB" excluded
snmp-server view "default" "snmpNotificationMIB" excluded
snmp-server view "default" "snmpUsmMIB" excluded
snmp-server view "default" "snmpVacmMIB" excluded
snmp-server view "default" "snmpCommunityMIB" excluded
snmp-server group "public" v1 read "default"
snmp-server group "public" v2c read "default"
snmp-server group "private" v1 read "default" write "default"
snmp-server group "private" v2c read "default" write "default"
snmp-server user "public" "public" v1
snmp-server user "private" "private" v1
snmp-server user "public" "public" v2c
```

```
snmp-server user "private" "private" v2c
snmp-server community-entry "Community1" "public" "public"
snmp-server community-entry "Community2" "private" "private"
!
cable modem offline aging-time 86400
!
bridge aging-time 15000
bridge 0
bridge 1
!
! no doxmonitor
!

file prompt alert
no cli logging
no cli logging password
cli logging path /
cli logging size 1024
alias scm "show cable modem"
!
clock timezone EST -5 0
!
no ip routing

default cpe subinterface Cable 1/0.0

default cm subinterface Cable 1/0.0
!
! attached sub-interfaces
!
interface FastEthernet 0/0
 ! description " "
 ! no shutdown
 ! mac-address 0000.ca3f.63ca
 duplex auto
 load-interval 300
 bridge-group 0
 ip address 10.1.176.240 255.255.255.192
 management-access
 no ip directed-broadcast
 no ip source-verify
 no ip source-verify subif
 no ip l2-bg-to-bg-routing
 ip verify-ip-address-filter
!
!
interface FastEthernet 0/1
 ! description " "
 ! no shutdown
 ! mac-address 0000.ca3f.63cb
 duplex auto
 load-interval 300
 bridge-group 0
 no management-access
 no ip directed-broadcast
 no ip source-verify
 no ip source-verify subif
 no ip l2-bg-to-bg-routing
 ip verify-ip-address-filter
!
!
interface Cable 1/0
 cable utilization-interval 10
 cable insertion-interval automatic
 cable sync-interval 10
 cable ucd-interval 2000
! cable max-sids 8192
 cable max-ranging-attempts 16
 cable sid-verify
 cable map-advance static
 cable downstream annex B
 cable downstream rate-limit token-bucket shaping auto-delay auto-value 80000
 cable flap-list size 500
 cable flap-list aging 259200
 cable flap-list miss-threshold 6
 cable flap-list insertion-time 180
! description " "
! no shutdown
! mac-address 0000.ca3f.63cc
load-interval 300
 cable downstream  load-interval 300
 bridge-group 0
 management-access
 l2-broadcast-echo
 l2-multicast-echo
 ip-broadcast-echo
 ip-multicast-echo
 ip igmp disable
 ip igmp version 2
 ip igmp robustness 2
 no ip igmp verify ip-router-alert-option
 no ip dhcp relay
 ip dhcp relay information option
 no ip dhcp relay validate renew
 cable helper-address 10.1.176.251
 cable dhcp-giaddr policy
 cable downstream channel-width 6mhz
 cable downstream frequency 681000000
 cable downstream interleave-depth 32
 cable downstream modulation 64qam
 cable downstream power-level 55
 cable privacy accept-self-signed-certificate
 no cable privacy check-cert-validity-periods
 cable privacy kek life-time 604800
 cable privacy tek life-time 43200
 no cable shared-secret
! no cable upstream 0 description
! no cable upstream 0 shutdown
 cable upstream 0  load-interval 300
 cable upstream 0 channel-type TDMA
 cable upstream 0 modulation-profile 1
 cable upstream 0 frequency 33000000
 no cable upstream 0 pre-equalization
 cable upstream 0 power-level 2 fixed
 cable upstream 0 channel-width 3200000
 cable upstream 0 group-id 1
 cable upstream 0 plant-length 160
 no cable upstream 0 ingress-cancellation
 cable upstream 0 periodic-maintenance-interval 1000
 cable upstream 0 short-periodic-maintenance-interval 100
 cable upstream 0 low-power-offset -60
 cable upstream 0 high-power-offset 60
 cable upstream 0 concatenation
 cable upstream 0 minislot-size 4
 cable upstream 0 trigger-index 0
 cable upstream 0 snr-timeconstant 9
 cable upstream 0 fragmentation
 cable upstream 0 rate-limit
 cable upstream 0 data-backoff 0 5
 cable upstream 0 range-backoff automatic
 cable upstream 0 status activate
! no cable upstream 1 description
! cable upstream 1 shutdown
 cable upstream 1  load-interval 300
 cable upstream 1 channel-type TDMA
 cable upstream 1 modulation-profile 1
 cable upstream 1 frequency 15000000
 no cable upstream 1 pre-equalization
 cable upstream 1 power-level -4 fixed
 cable upstream 1 channel-width 3200000
 cable upstream 1 group-id 2
 cable upstream 1 plant-length 160
 no cable upstream 1 ingress-cancellation
 cable upstream 1 periodic-maintenance-interval 1000
 cable upstream 1 short-periodic-maintenance-interval 100
 cable upstream 1 low-power-offset -60
 cable upstream 1 high-power-offset 60
 cable upstream 1 concatenation
 cable upstream 1 minislot-size 4
 cable upstream 1 trigger-index 0
 cable upstream 1 snr-timeconstant 9
 cable upstream 1 fragmentation
 cable upstream 1 rate-limit
 cable upstream 1 data-backoff 0 5
 cable upstream 1 range-backoff automatic

 no ip directed-broadcast
 no ip source-verify
 no ip source-verify subif
 no ip l2-bg-to-bg-routing
 ip verify-ip-address-filter
!
!
! unattached subinterfaces
!
!
interface FastEthernet 0/1.1
 no shutdown
 no management-access
 no ip directed-broadcast
 no ip source-verify
 no ip source-verify subif
 no ip l2-bg-to-bg-routing
 ip verify-ip-address-filter
!
!
interface Cable 1/0.1
 cable utilization-interval 10
 cable sid-verify
 no shutdown
 no management-access
 l2-broadcast-echo
 l2-multicast-echo
 ip-broadcast-echo
 ip-multicast-echo
 no ip dhcp relay
 no ip dhcp relay information option
 no ip dhcp relay validate renew
 no cable dhcp-giaddr
 no ip directed-broadcast
 no ip source-verify
 no ip source-verify subif
 no ip l2-bg-to-bg-routing
 ip verify-ip-address-filter
!

! Igmp Proxy configuration
!
!
```

```
key chain foo
!
!
ip default-gateway 10.1.176.254
!
!
!
!
cable modulation-profile 1 request AdvPhy TDMA&ATDMA 1 0
cable modulation-profile 1 request AdvPhy preamble-type qpsk0
cable modulation-profile 1 request 0 16 0 8 qpsk scrambler 338 no-diff 64 fixed
cable modulation-profile 1 initial AdvPhy TDMA&ATDMA 1 0
cable modulation-profile 1 initial AdvPhy preamble-type qpsk0
cable modulation-profile 1 initial 5 34 0 48 qpsk scrambler 338 no-diff 640
fixed
cable modulation-profile 1 station AdvPhy TDMA&ATDMA 1 0
cable modulation-profile 1 station AdvPhy preamble-type qpsk0
cable modulation-profile 1 station 5 34 0 48 qpsk scrambler 338 no-diff 384
fixed
cable modulation-profile 1 short AdvPhy TDMA
cable modulation-profile 1 short 6 78 13 8 qpsk scrambler 338 no-diff 84 fixed
cable modulation-profile 1 long AdvPhy TDMA
cable modulation-profile 1 long 8 220 0 8 qpsk scrambler 338 no-diff 96 fixed
cable modulation-profile 1 advPhyS AdvPhy ATDMA 1 1536
cable modulation-profile 1 advPhyS AdvPhy preamble-type qpsk1
cable modulation-profile 1 advPhyS 12 78 14 8 64qam scrambler 338 no-diff 104
fixed
cable modulation-profile 1 advPhyL AdvPhy ATDMA 1 1536
cable modulation-profile 1 advPhyL AdvPhy preamble-type qpsk1
cable modulation-profile 1 advPhyL 16 220 0 8 64qam scrambler 338 no-diff 104
fixed
cable modulation-profile 2 request AdvPhy TDMA&ATDMA 1 0
cable modulation-profile 2 request AdvPhy preamble-type qpsk0
cable modulation-profile 2 request 0 16 0 8 qpsk scrambler 338 no-diff 64 fixed
cable modulation-profile 2 initial AdvPhy TDMA&ATDMA 1 0
cable modulation-profile 2 initial AdvPhy preamble-type qpsk0
cable modulation-profile 2 initial 5 34 0 48 qpsk scrambler 338 no-diff 640
fixed
cable modulation-profile 2 station AdvPhy TDMA&ATDMA 1 0
cable modulation-profile 2 station AdvPhy preamble-type qpsk0
cable modulation-profile 2 station 5 34 0 48 qpsk scrambler 338 no-diff 384
fixed
cable modulation-profile 2 short AdvPhy TDMA
cable modulation-profile 2 short 6 78 7 8 16qam scrambler 338 no-diff 168 fixed
cable modulation-profile 2 long AdvPhy TDMA
cable modulation-profile 2 long 8 220 0 8 16qam scrambler 338 no-diff 192 fixed
cable modulation-profile 2 advPhyS AdvPhy ATDMA 1 1536
cable modulation-profile 2 advPhyS AdvPhy preamble-type qpsk1
cable modulation-profile 2 advPhyS 12 78 14 8 64qam scrambler 338 no-diff 104
fixed
cable modulation-profile 2 advPhyL AdvPhy ATDMA 1 1536
cable modulation-profile 2 advPhyL AdvPhy preamble-type qpsk1
cable modulation-profile 2 advPhyL 16 220 0 8 64qam scrambler 338 no-diff 104
fixed
!
cable frequency-band 1 1 start 1800000 stop 68200000
cable frequency-band 2 1 start 1800000 stop 68200000
cable frequency-band 3 1 start 1800000 stop 68200000
cable frequency-band 4 1 start 1800000 stop 68200000
cable frequency-band 5 1 start 1800000 stop 68200000
cable frequency-band 6 1 start 1800000 stop 68200000
!
no cable group 1 load-balancing
!no cable group 1 description
no cable group 2 load-balancing
!no cable group 2 description
no cable group 3 load-balancing
!no cable group 3 description
no cable group 4 load-balancing
!no cable group 4 description
no cable group 5 load-balancing
!no cable group 5 description
no cable group 6 load-balancing
!no cable group 6 description
!
!
!
MIB ifTable 1 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 2 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 3 up_ifAdmin Disable_ifLinkTrap " "
MIB ifTable 4 up_ifAdmin Enable_ifLinkTrap " "
MIB ifTable 5 up_ifAdmin Disable_ifLinkTrap ""
MIB ifTable 6 down_ifAdmin Disable_ifLinkTrap ""
MIB ifTable 11 up_ifAdmin Enable_ifLinkTrap ""
MIB ifTable 12 down_ifAdmin Enable_ifLinkTrap ""
!
logging syslog host 10.1.178.124
logging thresh none
logging thresh interval 1
logging severity 0 local trap sys no-vol
logging severity 1 local trap sys no-vol
logging severity 2 local trap sys no-vol
logging severity 3 local trap sys vol
logging severity 4 local trap sys vol
logging severity 5 local trap sys vol
logging severity 6 local trap sys no-vol
logging severity 7 local trap sys no-vol
logging trap-control 0x0
elog on
elog size 50
!
```

```
!cable service class "Multicast" priority  0
!cable service class "Multicast" sched-type best-effort
!cable service class "Multicast" downstream
!cable service class "Multicast" activity-timeout  0
!cable service class "Multicast" admission-timeout  0
!cable service class "Multicast" grant-interval  0
!cable service class "Multicast" grant-jitter  0
!cable service class "Multicast" grant-size  0
!cable service class "Multicast" grants-per-interval  0
!cable service class "Multicast" max-burst  0
!cable service class "Multicast" max-concat-burst  0
!cable service class "Multicast" max-latency  0
!cable service class "Multicast" max-rate  0
!cable service class "Multicast" min-packet-size  0
!cable service class "Multicast" min-rate  0
!cable service class "Multicast" poll-interval  0
!cable service class "Multicast" poll-jitter  0
!cable service class "Multicast" req-trans-policy  0x0
!cable service class "Multicast" tos-overwrite  0x0 0x0
!cable service class "Multicast" status activate
!
cable filter
cable submgmt
cable submgmt cpe ip filtering
no cable submgmt default active
cable submgmt default learnable
cable submgmt default max-cpe 16
cable submgmt default filter-group cm upstream 0
cable submgmt default filter-group cm downstream 0
cable submgmt default filter-group cpe upstream 0
cable submgmt default filter-group cpe downstream 0
!
!
line console
 length 24
 width 80
 timeout 900
 monitor
 no vt100-colours
line vty 0 0
 length 0
 width 80
 timeout 65000
 no monitor
 no vt100-colours
line vty 1 1
 length 42
 width 80
 timeout 65000
 no monitor
 no vt100-colours
line vty 2 2
 length 0
 width 80
 timeout 65000
 no monitor
 no vt100-colours
line vty 3 3
 length 0
 width 80
 timeout 65000
 no monitor
 no vt100-colours
!
no ipdr
ipdr filename "ipdr.xml.gz"
ipdr login "anonymous"
ipdr password "anonymous"
!
ntp server 129.6.15.28 interval 300
ntp server 129.6.15.28 master
!
!
exception auto-reboot 0
exception 3212-monitor reset
C3#
```

Default Modulation Profiles

The following are the default modulation profiles created with the cable modulation-profile command.

Default QPSK Profile

```
C3(config)#cable modulation-profile 2 qpsk
C3(config)#show cable modulation-profile

Mod IUC     Type Preamb Diff FEC    FEC    Scrambl Max  Guard Last Scrambl
                 length enco T      CW     Seed    B    time  CW
                             BYTES  SIZE           size size  short
2   request qpsk  64    no   0x0    0x10  0x152   0    8     no   yes
2   initial qpsk  640   no   0x5    0x22  0x152   0    48    no   yes
2   station qpsk  384   no   0x5    0x22  0x152   0    48    no   yes
2   short   qpsk  64    no   0x6    0x4b  0x152   14   8     no   yes
2   long    qpsk  64    no   0x8    0xdc  0x152   0    8     no   yes
2   advPhyS qpsk  104   no   0xc    0x4b  0x152   6    8     no   yes
2   advPhyL qpsk  104   no   0x10   0xdc  0x152   0    8     no   yes
2   advPhyU qpsk  104   no   0x10   0xdc  0x152   0    8     no   yes
```

Default QAM Profile

```
C3(config)#cable modulation-profile 2 qam
C3(config)#show cable modulation-profile

Mod IUC     Type Preamb Diff FEC    FEC    Scrambl Max  Guard Last Scrambl
                 length enco T      CW     Seed    B    time  CW
                             BYTES  SIZE           size size  short
2   request 16qam 64    no   0x0    0x10  0x152   0    8     no   yes
2   initial 16qam 640   no   0x5    0x22  0x152   0    48    no   yes
2   station 16qam 384   no   0x5    0x22  0x152   0    48    no   yes
2   short   16qam 64    no   0x6    0x4b  0x152   14   8     no   yes
2   long    16qam 64    no   0x8    0xdc  0x152   0    8     no   yes
2   advPhyS 16qam 104   no   0xc    0x4b  0x152   6    8     no   yes
2   advPhyL 16qam 104   no   0x10   0xdc  0x152   0    8     no   yes
2   advPhyU 16qam 104   no   0x10   0xdc  0x152   0    8     no   yes
```

Default Advanced PHY Profile

```
C3(config)#cable modulation-profile 2 advanced-phy
C3(config)#show cable modulation-profile

Mod IUC     Type Preamb Diff FEC   FEC  Scrambl Max  Guard Last Scrambl
                 length enco T     CW   Seed    B    time  CW
                             BYTES SIZE         size size  short
2   request qpsk  64    no   0x0   0x10 0x152   0    8     no   yes
2   initial qpsk  640   no   0x5   0x22 0x152   0    48    no   yes
2   station qpsk  384   no   0x5   0x22 0x152   0    48    no   yes
2   short   qpsk  64    no   0x6   0x4b 0x152   14   8     no   yes
2   long    qpsk  64    no   0x8   0xdc 0x152   0    8     no   yes
2   advPhyS 64qam 104   no   0xc   0x4b 0x152   6    8     no   yes
2   advPhyL 64qam 104   no   0x10  0xdc 0x152   0    8     no   yes
2   advPhyU 64qam 104   no   0x10  0xdc 0x152   0    8     no   yes
```

Default Mixed Profile

```
C3(config)#cable modulation-profile 2 mix
C3(config)#show cable modulation-profile

Mod IUC     Type Preamb Diff FEC   FEC  Scrambl Max  Guard Last Scrambl
                 length enco T     CW   Seed    B    time  CW
                             BYTES SIZE         size size  short
2   request qpsk  64    no   0x0   0x10 0x152   0    8     no   yes
2   initial qpsk  640   no   0x5   0x22 0x152   0    48    no   yes
2   station qpsk  384   no   0x5   0x22 0x152   0    48    no   yes
2   short   16qam 64    no   0x6   0x4b 0x152   14   8     no   yes
2   long    16qam 64    no   0x8   0xdc 0x152   0    8     no   yes
2   advPhyS 16qam 104   no   0xc   0x4b 0x152   6    8     no   yes
2   advPhyL 16qam 104   no   0x10  0xdc 0x152   0    8     no   yes
2   advPhyU 16qam 104   no   0x10  0xdc 0x152   0    8     no   yes
```

D Configuration Forms

Use the following forms to record information about how the CMTS should be configured.

Booting Configuration

Boot device	□ Compact Flash disk □ TFTP server
Image file name	
Booting interface	□ fastethernet 0/0 □ fastethernet 0/1

TFTP Server Boot Parameters

(required only if you are network booting)

CMTS IP Address	
Subnet mask	
Gateway IP address	
VLAN ID (if necessary)	

Running Configuration - IP Networking

Ethernet interfaces in use	□ fastethernet 0/0 □ fastethernet 0/1
Management interface and options	□ fastethernet 0/0 □ fastethernet 0/1
Management IP address	
Management Subnet mask	
Gateway IP address	
VLAN ID (if necessary)	

TFTP Server Parameters

IP Address	
Subnet mask	
Gateway address (if necessary)	
VLAN ID (if necessary)	

DHCP Server 1 Parameters

IP Address	
Subnet mask	
Gateway address (if necessary)	
VLAN ID (if necessary)	

DHCP Server 2 Parameters

IP Address	
Subnet mask	
Gateway address (if necessary)	
VLAN ID (if necessary)	

DHCP Server 3 Parameters

IP Address	

Subnet mask	
Gateway address (if necessary)	
VLAN ID (if necessary)	

Fastethernet 0/0 Configuration

Physical Interface Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 1 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 2 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 3 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 4 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 5 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 6 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 7 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 8 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Fastethernet 0/1 Configuration

Physical Interface Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 1 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 2 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 3 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 4 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 5 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 6 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 7 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Sub-interface 8 Configuration

IP Address	
Subnet mask	
VLAN ID (if necessary)	

Cable Configuration

IP Networking

Make additional copies of this checklist for each sub-interface.

Helper Address 1	
	□ for modems □ for hosts
Helper Address 2	
	□ for modems □ for hosts
Helper Address 3	
	□ for modems □ for hosts
Helper Address 4	
	□ for modems □ for hosts

Helper Address 5	
	□ for modems □ for hosts
dhcp-giaddr	□ primary □ policy
Other DHCP options	□ ip dhcp relay □ ip dhcp relay information option

Downstream RF Configuration

DOCSIS type	□ DOCSIS (6 MHz) □ EuroDOCSIS (8 MHz)
Center Frequency (MHz)	
Modulation	□ 64 QAM □ 256 QAM

Upstream 0 RF Configuration

Center Frequency (MHz)	
Channel Width (MHz)	
Modulation	□ QPSK □ 8 QAM □ 16 QAM □ 32 QAM □ 64 QAM

Upstream 1 RF Configuration

Center Frequency (MHz)	
Channel Width (MHz)	
Modulation	□ QPSK □ 8 QAM □ 16 QAM □ 32 QAM □ 64 QAM

Upstream 2 RF Configuration

Center Frequency (MHz)	
Channel Width (MHz)	
Modulation	□ QPSK □ 8 QAM □ 16 QAM □ 32 QAM □ 64 QAM

Upstream 3 RF Configuration

Center Frequency (MHz)	
Channel Width (MHz)	
Modulation	□ QPSK □ 8 QAM □ 16 QAM □ 32 QAM □ 64 QAM

Upstream 4 RF Configuration

Center Frequency (MHz)	
Channel Width (MHz)	
Modulation	□ QPSK □ 8 QAM □ 16 QAM □ 32 QAM □ 64 QAM

Upstream 5 RF Configuration

Center Frequency (MHz)	
Channel Width (MHz)	
Modulation	□ QPSK □ 8 QAM □ 16 QAM □ 32 QAM □ 64 QAM

E Glossary

The following is a list of terms and abbreviations used in this manual.

Terminology

broadband

Transmission system that combines multiple independent signals onto one cable. In the cable industry, broadband refers to the frequency-division multiplexing of many signals in a wide bandwidth of RF frequencies using a hybrid fiber-coaxial (HFC) network.

carrier

A signal on which another, lower-frequency signal is modulated in order to transport the lower-frequency signal to another location.

Carrier-to-Noise C/N (also CNR)

The difference in amplitude between the desired RF carrier and the noise in a portion of the spectrum.

CATV

Acronym for community antenna television or cable television. Now refers to any coaxial or fiber cable-based system that provides television services.

channel

A specific frequency allocation and bandwidth. Downstream channels used for television are 6 MHz wide in the United States and 8 MHz wide in Europe.

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.

CM

Cable Modem. Typically a device installed at the subscriber premises that provides a high-speed data (Internet) connection through the HFC network.

CMTS

Cable Modem Termination System. A device at a cable headend that connects to cable modems over an HFC network to an IP network.

coaxial cable

The principal physical media over which CATV systems are built.

CPE

Customer Premises Equipment. Subscriber-owned equipment connected to the network. Technically, a cable modem, MTA, or NIU falls into this category, although many operators do not designate them as such.

CVC

Code Verification Certificate. A digital certificate containing a public key used to verify an encrypted software load downloaded to a cable modem. The manufacturer uses a private key to sign the image; the cable modem uses the public key contained in the CVC to verify the image.

dB

Decibel. A measure of the relative strength of two signals.

dBm

Decibels with respect to one milliwatt. A unit of RF signal strength used in satellite work and other communications applications.

dBmV

Decibels with respect to one millivolt in a 75-ohm system. This is the unit of RF power used in CATV work in North America: dBmV=dBm–48.75.

DHCP

Dynamic Host Configuration Protocol. An IP protocol used to provide an IP address and location of services (such as DNS and TFTP) needed by a device connecting to the network.

DNS

Domain Name Service (Server). An IP service that associates a domain name (such as www.example.com) with an IP address.

Downstream

In an HFC network, the direction from the headend to the subscriber. Some older cable documentation may refer to this as the forward path.

DOCSIS

Data Over Cable Service Interface Specification. The interoperability standards used for data communications equipment on an HFC network.

EuroDOCSIS

The European version of DOCSIS. EuroDOCSIS specifies an 8MHz downstream bandwidth (vs. 6MHz for DOCSIS); other minor differences exist as well.

FDM

Frequency Division Multiplexing. A data transmission method in which a number of transmitters share a transmission medium, each occupying a different frequency.

FEC

Forward Error Correction. In data transmission, a process by which additional data is added that is derived from the payload by an assigned algorithm. It allows the receiver to determine if certain classes of errors have occurred in transmission and, in some cases, allows other classes of errors to be corrected.

FQDN

Fully Qualified Domain Name. The name used to identify a single device on the Internet. See RFC821 for details.

Headend

The “central office” in an HFC network. The headend houses both video and data equipment. In larger MSO networks, a “master” headend often feeds several “remote” headends to provide distributed services.

HFC

Hybrid Fiber-Coaxial. A broadband, bi-directional shared media transmission system using fiber trunks between the headend and fiber nodes, and coaxial distribution cable between the fiber nodes and subscriber premises.

host

Any end-user computer system that connects to a network. In this document, the term host refers to the computer system connected to the LAN interface of the cable access router.

ingress noise

Over-the-air signals that are inadvertently coupled into the nominally closed coaxial cable distribution system. Ingress noise is difficult to track down and intermittent in nature.

MAC layer

Media Access Control sublayer. Controls access by the cable access router to the CMTS and to the upstream data slots.

MCNS

Multimedia Cable Network System Partners, Ltd. A consortium of cable companies providing service to the majority of homes in the United States and Canada. This consortium has decided to drive a standard with the goal of having interoperable cable access routers.

Maintenance window

The usual period of time for performing maintenance and repair operations. Since these activities often affect service to one or more subscribers, the maintenance window is usually an overnight period (often 1 a.m. to 5 a.m. local time).

MD5

Message Digest 5. A one-way hashing algorithm that maps variable length plaintext into fixed-length (16-byte) ciphertext. MD5 files, built by a provisioning server, contain provisioning data for each cable modem or NIU on the network.

MIB

Management Information Base. The data representing the state of a managed object in an SNMP-based network management system. Often used colloquially to refer to a single object or variable in the base; e.g. “the lcCmtsUpMaxCbrFlows MIB.”

MSO

Multi-System Operator. A cable company that operates multiple headend locations, usually in several cities.

narrowband

A single RF frequency.

NIU

Network Interface Unit. Used in this document as a generic term for a cable modem.

NMS

Network Management System. Software, usually SNMP-based, that allows you to monitor and control devices on the network. In a ToIP network, managed devices include cable modems, NIUs, CMTS, servers, PSTN interface devices, and routers. An NMS works by reading and setting values of MIB variables presented by each device.

NTSC

National Television Systems Committee. A United States TV technical standard, named after the organization that created the standard in 1941. Specifies a 6 MHz-wide modulated signal.

QAM

Quadrature Amplitude Modulation. A method of modulating digital signals onto an RF carrier, involving both amplitude and phase coding. QAM16 modulation encodes four digital bits per state and is used on upstream carriers; QAM64 and QAM256 encode six or eight bits (respectively) for use on downstream carriers.

QPSK

Quadrature Phase Shift Keying. A method of modulating digital signals onto an RF carrier, using four phase states to encode two digital bits.

ranging

The process of acquiring the correct timing offset such that the transmissions of a cable access router are aligned with the correct mini-slot boundary.

RF

Radio Frequency.

SID (Service Identifier)

A number that defines (at the MAC sublayer) a particular mapping between a cable access router (CM) and the CMTS. The SID is used for the purpose of upstream bandwidth allocation and class-of-service management.

Signal-to-Noise Ratio (SNR)

The difference in amplitude between a baseband signal and the noise in a portion of the spectrum.

SNMP

Simple Network Management Protocol.

symbol

Phase range of a sine wave.

tap

A device installed in the feeder cable that connects the home TV set to the cable network. Also called a drop.

TFTP

Trivial File Transfer Protocol. Used in DOCSIS networks to transfer software and provisioning files to network devices.

Upstream

The path from a subscriber device to the headend. Some older cable documentation may refer to this as the return path or reverse path.

F Index

Cadant C3 CMTS

Installation, Operation, and Maintenance Guide

All information contained in this document is subject to change without notice. Arris Interactive reserves the right to make changes to equipment design or program components, as progress in engineering, manufacturing methods, or other circumstances may warrant.

Document number: ARSVD00814
Release 3.0 Standard 2.0
March 2004

ARRIS™