Troubleshooting Tips for the Cisco uBR904 Cable Modem

The following sections are provided:

Feature Summary

This document provides MSOs (multiple service operators) with a set of software tools for troubleshooting a cable modem for data-over-cable connections. These tools are Cisco IOS troubleshooting commands used for verifying communication between the cable modem and other peripheral devices installed in the HFC network, such as the headend Cisco uBR7246, a DHCP server, and TFTP server.

Benefits

- A MAC layer system log file exists inside the cable modem, which provides you with a snapshot of detailed reasons why an interface might reset and all the negotiations that occurred between the cable modem and the CMTS (a Cisco uBR7246 positioned at the headend). Over 220 possible description fields exist in this log. The log is displayed using the **show controller cable-modem 0 mac log** EXEC command.
- Debug does not need to be turned on to troubleshoot a cable modem.
- Cable technicians can understand the progression of normal data-over-cable communication events, from which they can resolve faulty system connections.
- A cable technician can remotely telnet into a Cisco uBR904 cable modem, which could be installed in a customer's home, and perform simple diagnostic tasks.

List of Terms

CATV—Originally stood for Community Antenna Television. CATV now refers to any cable (coaxial/fiber) based system that provides television services.

Cable modem—Any device that modulates and demodulates digital data onto a CATV plant.

Cable router—A modular chassis-based router optimized for the data over CATV HFC applications.

Channel—A specific frequency allocation and bandwidth. Downstream channels used for television in the U. S. are 6 MHz wide.

CM—Cable modem.

CMTS—Cable Modem Termination System. Any DOCSIS compliant headend cable router, such as the Cisco uBR7246.

Downstream—The set of frequencies used to send data from a headend to a subscriber.

Headend—Central distribution point for a CATV system. Video signals are received here from satellite (either co-located or remote), frequency converted to the appropriate channels, combined with locally originated signals, and rebroadcast onto the HFC plant. For a CATV data system, the headend is the typical place to link between the HFC system and any external data networks.

HFC—Hybrid fiber-coaxial (cable). Older CATV systems were provisioned using only coaxial cable. Modern systems use fiber transport from the headend to an optical node located in neighborhood to reduce system noise. Coax runs from the node to the subscriber. The fiber plant is generally a star configuration with all optical node fibers terminating at a headend. The coaxial cable part of the system is generally a trunk-and-branch configuration.

Host—Any end-user computer system that connects to a network. The term host here refers to computer systems connected to the LAN interface of the cable modem.

MAC layer—Media Access Control sublayer. Controls access by the cable modem to the CMTS and to the upstream data slots.

MCNS—Multimedia Cable Network System Partners Ltd., a consortium of cable companies representing the majority of homes in the U.S. and Canada who have decided to derive a standard with the goal of having interoperable cable modems.

MSO—Multiple System Operator

QAM—Modulation scheme mostly used in the downstream direction (QAM-64, QAM-256). QAM-16 is expected to be usable in the upstream direction. Numbers indicate number of code points per symbol. The QAM rate or the number of points in the QAM constellation can be computed by 2 raised to the power of <number of bits/symbol>.

QPSK—Modulation scheme used in the upstream direction. Supports two data bits per symbol.

Subscriber Unit (SU)—An alternate term for cable modem. See *cable modem.*

Upstream—The set of frequencies used to send data from a subscriber to the headend.

Prerequisites

Cisco IOS Release 11.3(4)NA or later must be running inside the cable modem. Enter the **show version** EXEC command to display the software version level.

Supported MIBs and RFCs

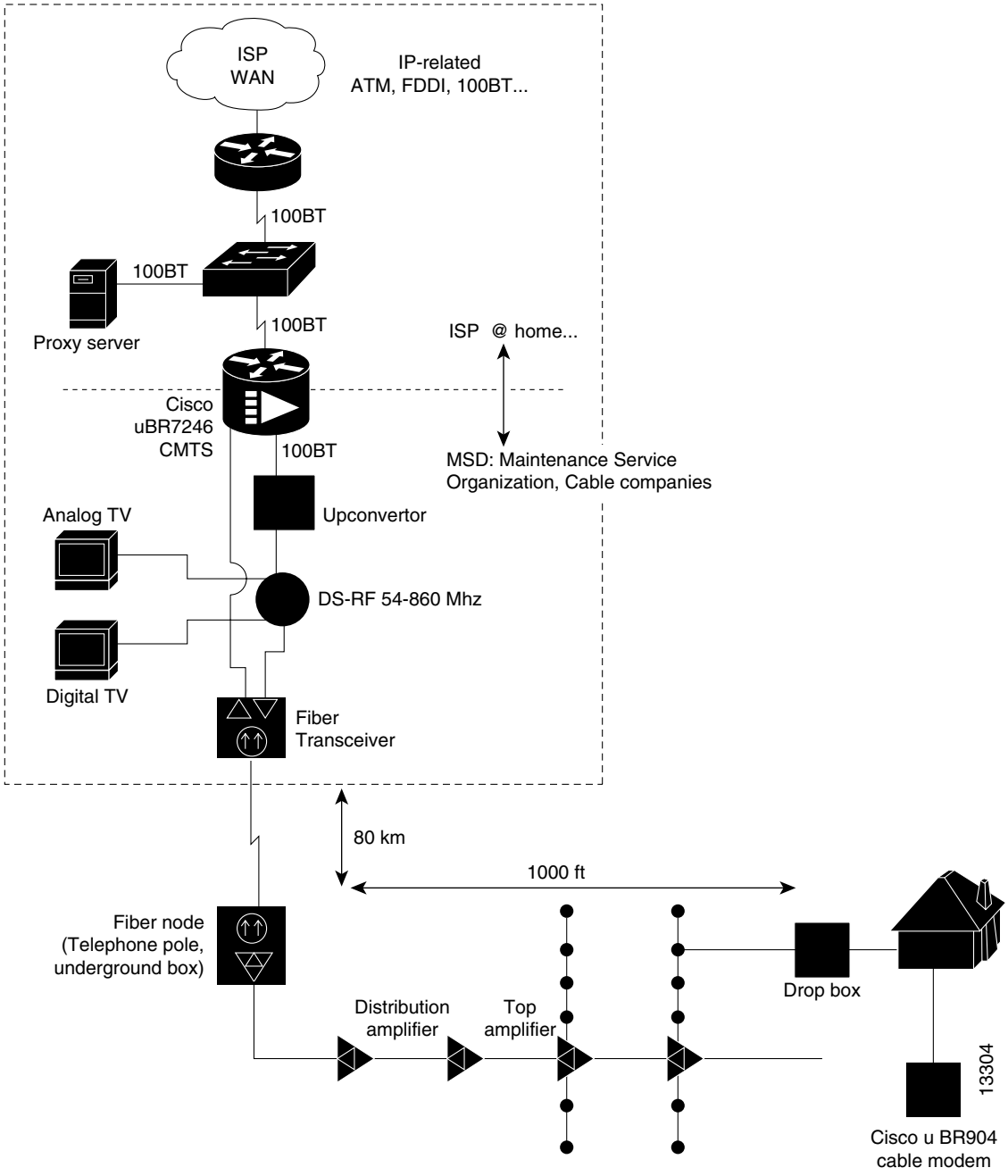
The Cisco uBR904 cable modem supports the following:

- Radio Frequency Interface Specification—This specification is developed by the Multimedia Cable Network System (MCNS) corsortium. It defines the radio-frequency interface specification for high-speed data-over-cable systems.
- Cisco Standard MIBs—The Cisco Standard MIBs consist of CiscoWorks. CiscoWorks is the network management program for planning, troubleshooting, and monitoring Cisco internetworks. CiscoWorks uses SNMP protocols to monitor all SNMP devices.
  - For more information about CiscoWorks on CCO, follow this path: **Products & Ordering: Cisco Products: Network Management: CiscoWorks**
  - For more information about CiscoWorks on the Documentation CD-ROM, follow this path: **Cisco Product Documentation: Network Management: CiscoWorks**
- RF Interface MIB—The Radio Frequency Interface (RFI) MIB is specific to cable Data Over Cable Interface Specification (DOCSIS) implementations. The RIF MIB provides an interface that permits management of the Cisco uBR904 cable modem over the cable or Ethernet interface. Using SNMP management applications, this MIB allows access to statistics such as MAC, driver configuration, and counters through its interface.
- Cable Device MIB—The Cable Device MIB records statistics related to the configuration and status of the Cisco uBR904 cable modem. Statistics include an events log and device status. The Cable Device MIB is very similar to the RFI MIB in that both allow access to statistics; they are different in that the Cable Device MIB reports statistics on the Cisco uBR904 cable modem, while the RFI MIB reports statistics on the radio frequency transmissions over the cable television line.

CMTS to Cable Modem Network Topology

Figure 1 shows the physical relationship between the devices in the HFC network and the cable modem.

Figure 1 Sample Topology



Troubleshooting Tips

Perform the following steps to troubleshoot a cable modem:

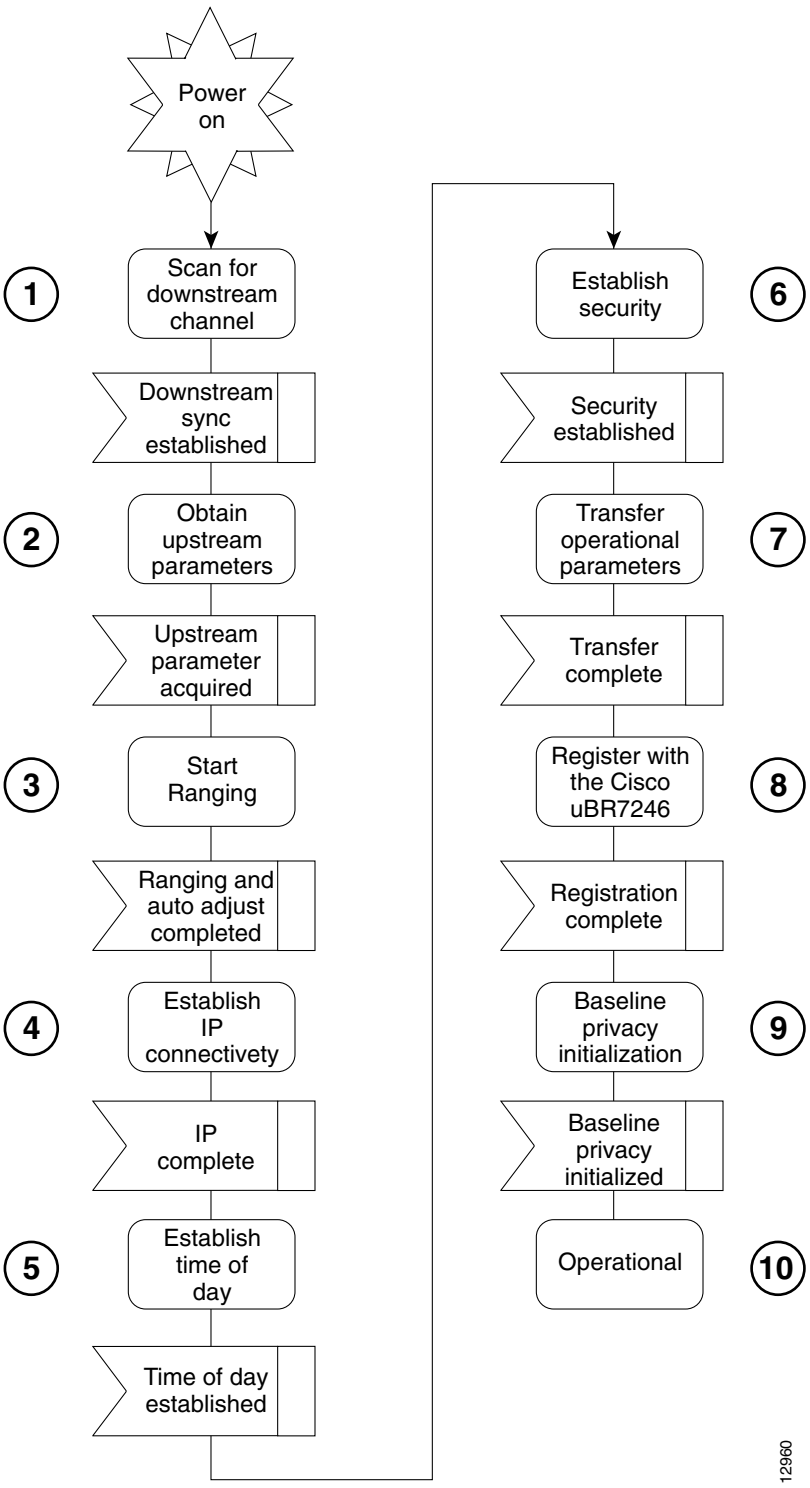
- Step 1—Understand How Basic Initialization Works
- Step 2—Connect to the Cable Modem
- Step 3—Display the Cable Modem's MAC Log File
- Step 4—Interpret the MAC Log File and Take Action
- Step 5—(Optional) Use Additional Troubleshooting Commands

Understand How Basic Initialization Works

Before you troubleshoot a cable modem, you should be familiar with the cable modem initialization process. See Figure 2 and Table 1. Understanding this flowchart and sequence of events helps you determine where and why connections fail.

The sequence numbers shown in Figure 2 are explained in Table 1, which appears after the illustration. The cable modem will complete all the steps in this flowchart each time the cable modem needs to reestablish ranging and registration with the CMTS.

Figure 2 Cable Modem Initialization Flowchart



.

Table 1 Cable Modem Initialization Sequences and Events

Sequence	Event	Description
1	Scan for a downstream channel and establish synchronization with the headend Cisco uBR7246.	The cable modem acquires a downstream channel from the headend, saves the last operational frequency in non-volatile memory, and tries to reacquire the saved downstream channel the next time a request is made. An ideal downstream signal is one that synchronizes QAM symbol timing, FEC framing, MPEG packetization, and recognizes downstream sync MAC layer messages.
2	Obtain upsteam channel parameters.	The cable modem waits for an upstream channel descriptor messages (UCD) from the headend Cisco uBR7246. This is done to retrieve transmission parameters for the upstream channel.
3	Start ranging for power adjustments.	The ranging process adjusts cable modem's transmit power. The cable modem performs ranging in stages, ranging state 1 and ranging state 2.
4	Establish IP connectivity.	The cable modem invokes DHCP requests to obtain an IP address, which is needed for IP connectivity. The DHCP request also includes the name of a file that contains additional configuration parameters, the TFTP server's address, and the Time of Day (TOD) server's address.
5	Establish the time of day.	The Cisco uBR904 cable modem accesses the TOD server for the current date and time, which is used to create time stamps for logged events (such as those displayed in the MAC log file).
6	Establish security.	Keys for privacy are exchanged between the cable modem and the CMTS (the Cisco uBR7246 cable router).
7	Transfer operational parameters.	After the DHCP and security operations are successful, the cable modem downloads operational parameters from a configuration file stored on a cable company's TFTP server.
8	Perform registration.	The cable modem registers with the headend Cisco uBR7246. The cable modem is authorized to forward traffic into the cable network after the cable modem is initialized, authenticated, and configured. **Note** The Cisco uBR904 cable modem supports baseline privacy in Cisco IOS Release 11.3(5)NA and later.
9	Comply with baseline privacy.	Link level encryption keys are exchanged between the headend and the cable modem.
10	Enter the operational maintenance state.	As soon as the cable modem is completely up and running, it enters operational maintenance state.

Connect to the Cable Modem

Telnet to the IP address assigned to the cable interface or Ethernet interface. If the interface is not up, you need to access the Cisco IOS software via the RJ-45 console port, which is a physical port on the back of the cable modem.

Because the MAC log file only holds a snapshot of 1023 entries at a time, you should try to display the cable modem's log file within 5 minutes of when the reset or problem occurs.

Display the Cable Modem's MAC Log File

A MAC layer circular log file is inside the cable modem. This file contains a history of log messages, such as state event activities and timestamps, which are used for troubleshooting purposes. This log file contains the most valuable information for a cable technician or engineer to debug the cable interface.

The MAC log file is displayed by entering the **show controller cable-modem 0 mac log** EXEC command.

The most useful display fields in this log file are the reported state changes. These fields are preceded by the message `CMAC_LOG_STATE_CHANGE`. These fields show how the cable modem progresses through the various processes involved in establishing communication with the CMTS and registration. The `maintenance_state` is the normal operational state, and the `wait_for_link_up_state` is the normal state when the interface is shutdown.

Here is the normal progression of states as displayed by the MAC log:

```
wait_for_link_up_state
ds_channel_scanning_state
wait_ucd_state
wait_map_state
ranging_1_state
ranging_2_state
dhcp_state
establish_tod_state
security_association_state
configuration_file_state
registration_state
establish_privacy_state
maintenance_state
```

Note To translate this output into more meaningful information, see the "Interpret the MAC Log File and Take Action" section on page 10.

Here is what an example MAC log file looks like when the cable modem interface successfully comes up and registers with the cable network. The output you see is directly related to the messages that are exchanged between the cable modem and the CMTS (the Cisco uBR7246).

```
uBR904# show controller cable-modem 0 mac log
508144.340 CMAC_LOG_DRIVER_INIT_IDB_RESET              0x08098FEA
508144.342 CMAC_LOG_LINK_DOWN
508144.344 CMAC_LOG_LINK_UP
508144.348 CMAC_LOG_STATE_CHANGE                       ds_channel_scanning_state
508144.350 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      88/453000000/855000000/6000000
508144.354 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      89/93000000/105000000/6000000
508144.356 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      90/111250000/117250000/6000000
508144.360 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      91/231012500/327012500/6000000
508144.362 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      92/333015000/333015000/6000000
508144.366 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      93/339012500/399012500/6000000
508144.370 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      94/405000000/447000000/6000000
508144.372 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      95/123015000/129015000/6000000
508144.376 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      96/135012500/135012500/6000000
508144.380 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      97/141000000/171000000/6000000
508144.382 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      98/219000000/225000000/6000000
508144.386 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      99/177000000/213000000/6000000
508144.390 CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY     699000000
508145.540 CMAC_LOG_UCD_MSG_RCVD                       3
508146.120 CMAC_LOG_DS_64QAM_LOCK_ACQUIRED             699000000
```

```
508146.122 CMAC_LOG_DS_CHANNEL_SCAN_COMPLETED
508146.124 CMAC_LOG_STATE_CHANGE                        wait_ucd_state
508147.554 CMAC_LOG_UCD_MSG_RCVD                        3
508147.558 CMAC_LOG_UCD_NEW_US_FREQUENCY                20000000
508147.558 CMAC_LOG_SLOT_SIZE_CHANGED                   8
508147.622 CMAC_LOG_FOUND_US_CHANNEL                    1
508147.624 CMAC_LOG_STATE_CHANGE                        wait_map_state
508148.058 CMAC_LOG_MAP_MSG_RCVD
508148.060 CMAC_LOG_INITIAL_RANGING_MINISLOTS           40
508148.062 CMAC_LOG_STATE_CHANGE                        ranging_1_state
508148.064 CMAC_LOG_RANGING_OFFSET_SET_TO               9610
508148.066 CMAC_LOG_POWER_LEVEL_IS                      28.0  dBmV (commanded)
508148.068 CMAC_LOG_STARTING_RANGING
508148.070 CMAC_LOG_RANGING_BACKOFF_SET                 0
508148.072 CMAC_LOG_RNG_REQ_QUEUED                      0
508148.562 CMAC_LOG_RNG_REQ_TRANSMITTED
508148.566 CMAC_LOG_RNG_RSP_MSG_RCVD
508148.568 CMAC_LOG_RNG_RSP_SID_ASSIGNED                2
508148.570 CMAC_LOG_ADJUST_RANGING_OFFSET               2408
508148.572 CMAC_LOG_RANGING_OFFSET_SET_TO               12018
508148.574 CMAC_LOG_ADJUST_TX_POWER                     20
508148.576 CMAC_LOG_POWER_LEVEL_IS                      33.0  dBmV (commanded)
508148.578 CMAC_LOG_STATE_CHANGE                        ranging_2_state
508148.580 CMAC_LOG_RNG_REQ_QUEUED                      2
508155.820 CMAC_LOG_RNG_REQ_TRANSMITTED
508155.824 CMAC_LOG_RNG_RSP_MSG_RCVD
508155.826 CMAC_LOG_ADJUST_RANGING_OFFSET               -64
508155.826 CMAC_LOG_RANGING_OFFSET_SET_TO               11954
508155.828 CMAC_LOG_RANGING_CONTINUE
508165.892 CMAC_LOG_RNG_REQ_TRANSMITTED
508165.894 CMAC_LOG_RNG_RSP_MSG_RCVD
508165.896 CMAC_LOG_ADJUST_TX_POWER                     -9
508165.898 CMAC_LOG_POWER_LEVEL_IS                      31.0  dBmV (commanded)
508165.900 CMAC_LOG_RANGING_CONTINUE
508175.962 CMAC_LOG_RNG_REQ_TRANSMITTED
508175.964 CMAC_LOG_RNG_RSP_MSG_RCVD
508175.966 CMAC_LOG_RANGING_SUCCESS
508175.968 CMAC_LOG_STATE_CHANGE                        dhcp_state
508176.982 CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS            188.188.1.62
508176.984 CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS            4.0.0.1
508176.986 CMAC_LOG_DHCP_TOD_SERVER_ADDRESS             4.0.0.32
508176.988 CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
508176.988 CMAC_LOG_DHCP_TZ_OFFSET                      360
508176.990 CMAC_LOG_DHCP_CONFIG_FILE_NAME               platinum.cm
508176.992 CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
508176.996 CMAC_LOG_DHCP_COMPLETE
508177.120 CMAC_LOG_STATE_CHANGE                        establish_tod_state
508177.126 CMAC_LOG_TOD_REQUEST_SENT
508177.154 CMAC_LOG_TOD_REPLY_RECEIVED                  3107617539
508177.158 CMAC_LOG_TOD_COMPLETE
508177.160 CMAC_LOG_STATE_CHANGE                        security_association_state
508177.162 CMAC_LOG_SECURITY_BYPASSED
508177.164 CMAC_LOG_STATE_CHANGE                        configuration_file_state
508177.166 CMAC_LOG_LOADING_CONFIG_FILE                 platinum.cm
508178.280 CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
508178.300 CMAC_LOG_STATE_CHANGE                        registration_state
508178.302 CMAC_LOG_REG_REQ_MSG_QUEUED
508178.306 CMAC_LOG_REG_REQ_TRANSMITTED
508178.310 CMAC_LOG_REG_RSP_MSG_RCVD
508178.312 CMAC_LOG_COS_ASSIGNED_SID                    1/2
508178.314 CMAC_LOG_RNG_REQ_QUEUED                      2
508178.316 CMAC_LOG_REGISTRATION_OK
508178.318 CMAC_LOG_STATE_CHANGE                        establish_privacy_state
508178.320 CMAC_LOG_NO_PRIVACY
508178.322 CMAC_LOG_STATE_CHANGE                        maintenance_state
```

You can display other aspects of the MAC layer by using variations of the **show controller cable-modem 0 mac** command:

```
uBR904# show controller cable-modem 0 mac ?
  errors    Mac Error Log data
  hardware  All CM Mac Hardware registers
  log       Mac log data
  resets    Resets of the MAC
  state     Current MAC state
```

For examples and descriptions on how to use these keywords, see the **show controller cable-modem mac** command reference page.

Interpret the MAC Log File and Take Action

The MAC log file explains a detailed history of initialization events that occurred in the cable modem. All pertinent troubleshooting information is stored here.

The following sample log file is organized by chronological sequence event. Sample comments are also included.

- Event 1—Wait for the Link to Come Up
- Event 2—Scan for a Downstream Channel then Synchronize
- Event 3—Obtain Upstream Parameters
- Event 4—Start Ranging for Power Adjustments
- Event 5—Establish IP Connectivity
- Event 6—Establish the Time of Day
- Event 7—Establish Security
- Event 8—Transfer Operational Parameters
- Event 9—Perform Registration
- Event 10—Comply with Baseline Privacy
- Event 11—Enter the Maintenance State

Event 1—Wait for the Link to Come Up

The MAC layer informs the cable modem's drivers that it needs to reset. This is the first event that happens after the modem powers up and begins initialization. The fields `LINK_DOWN` and `LINK_UP` are similar to a shut and no shut on a standard Cisco interface.

```
uBR904# show controller cable-modem 0 mac log

528302.040 CMAC_LOG_LINK_DOWN
528302.042 CMAC_LOG_RESET_FROM_DRIVER
528302.044 CMAC_LOG_STATE_CHANGE                    wait_for_link_up_state
528302.046 CMAC_LOG_DRIVER_INIT_IDB_SHUTDOWN        0x08098D02
528302.048 CMAC_LOG_LINK_DOWN
528308.428 CMAC_LOG_DRIVER_INIT_IDB_RESET           0x08098E5E
528308.432 CMAC_LOG_LINK_DOWN
528308.434 CMAC_LOG_LINK_UP
```

Event 2—Scan for a Downstream Channel then Synchronize

Different geographical regions and different cable plants use different frequency bands. The Cisco uBR904 cable modem uses a built-in default frequency scanning feature to address this issue. After the cable modem finds a successful downstream frequency channel, it saves the channel to NVRAM. The cable modem recalls this value the next time it needs to synchronize its frequency.

The field CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND tells you what frequency the cable modem will scan for. The field CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY tells you the frequency the cable modem locked onto and saved to NVRAM (for future recall). The field CMAC_LOG_DS_64QAM_LOCK_ACQUIRED communicates the same information. The field CMAC_LOG_DS_CHANNEL_SCAN_COMPLETED indicates that the scanning and synchronization was successful.

```
508144.348 CMAC_LOG_STATE_CHANGE                        ds_channel_scanning_state
508144.350 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       88/453000000/855000000/6000000
508144.354 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       89/93000000/105000000/6000000
508144.356 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       90/111250000/117250000/6000000
508144.360 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       91/231012500/327012500/6000000
508144.362 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       92/333015000/333015000/6000000
508144.366 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       93/339012500/399012500/6000000
508144.370 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       94/405000000/447000000/6000000
508144.372 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       95/123015000/129015000/6000000
508144.376 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       96/135012500/135012500/6000000
508144.380 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       97/141000000/171000000/6000000
508144.382 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       98/219000000/225000000/6000000
508144.386 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND       99/177000000/213000000/6000000
508144.390 CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY      699000000
508145.540 CMAC_LOG_UCD_MSG_RCVD                        3
508146.120 CMAC_LOG_DS_64QAM_LOCK_ACQUIRED              699000000
508146.122 CMAC_LOG_DS_CHANNEL_SCAN_COMPLETED
```

A frequency band is a group of adjacent channels. These bands are numbered from 88 to 99. Each band has starting and ending digital carrier frequencies and a 6 MHz step size. For example, a search of EIA channels 95-97 is specified using band 89. The starting frequency is 93 Mhz, the ending frequency is 105 Mhz.

The cable modem's default frequency bands correspond to the North American EIA CATV channel plan for 6 MHz channel slots between 90 MHz and 858 MHz. For example, EIA channel 95 occupies the slot 90-96 MHz. The digital carrier frequency is specified as the center frequency of 93 MHz. Channel 95 is usually specified using the analog video carrier frequency of 91.25 Mhz, which lies 1.75 Mhz below the center of the slot.

The search table is arranged so that the first frequencies tried are above 450 Mhz. Because many CATV systems have been upgraded from 450 MHz to 750 MHz coaxial cable, digital channels have a high chance of being assigned in the new spectrum. The search table omits channels below 90 MHz and above 860 MHz since the DOCSIS specification does not mandate their coverage.

Some CATV systems use alternative frequency plans such as the IRC (Incrementally Related Carrier) and HRC (Harmonically Related Carrier) plans. Most of the IRC channel slots overlap the EIA plan. The HRC plan is not supported by Cisco's cable modems since so few cable plants are using this plan.

Event 3—Obtain Upstream Parameters

The cable modem waits for an upstream channel descriptor messages (UCD) from the headend Cisco uBR7246. This is done to retrieve transmission parameters for the upstream channel.

```
508146.124 CMAC_LOG_STATE_CHANGE                    wait_ucd_state
508147.554 CMAC_LOG_UCD_MSG_RCVD                    3
508147.558 CMAC_LOG_UCD_NEW_US_FREQUENCY            20000000
508147.558 CMAC_LOG_SLOT_SIZE_CHANGED               8
508147.622 CMAC_LOG_FOUND_US_CHANNEL                1
508147.624 CMAC_LOG_STATE_CHANGE                    wait_map_state
508148.058 CMAC_LOG_MAP_MSG_RCVD
508148.060 CMAC_LOG_INITIAL_RANGING_MINISLOTS       40
```

Event 4—Start Ranging for Power Adjustments

The ranging process adjusts cable modem's transmit power. The cable modem performs ranging in two stages, ranging state 1 and ranging state 2.

The field `CMAC_LOG_POWER_LEVEL_IS` is the power level that the Cisco uBR7246 told the cable modem to adjust to. The field `CMAC_LOG_RANGING_SUCCESS` indicates that the ranging adjustment was successful.

```
508148.062 CMAC_LOG_STATE_CHANGE                    ranging_1_state
508148.064 CMAC_LOG_RANGING_OFFSET_SET_TO           9610
508148.066 CMAC_LOG_POWER_LEVEL_IS                  28.0  dBmV (commanded)
508148.068 CMAC_LOG_STARTING_RANGING
508148.070 CMAC_LOG_RANGING_BACKOFF_SET             0
508148.072 CMAC_LOG_RNG_REQ_QUEUED                  0
508148.562 CMAC_LOG_RNG_REQ_TRANSMITTED
508148.566 CMAC_LOG_RNG_RSP_MSG_RCVD
508148.568 CMAC_LOG_RNG_RSP_SID_ASSIGNED            2
508148.570 CMAC_LOG_ADJUST_RANGING_OFFSET           2408
508148.572 CMAC_LOG_RANGING_OFFSET_SET_TO           12018
508148.574 CMAC_LOG_ADJUST_TX_POWER                 20
508148.576 CMAC_LOG_POWER_LEVEL_IS                  33.0  dBmV (commanded)
508148.578 CMAC_LOG_STATE_CHANGE                    ranging_2_state
508148.580 CMAC_LOG_RNG_REQ_QUEUED                  2
508155.820 CMAC_LOG_RNG_REQ_TRANSMITTED
508155.824 CMAC_LOG_RNG_RSP_MSG_RCVD
508155.826 CMAC_LOG_ADJUST_RANGING_OFFSET           -64
508155.826 CMAC_LOG_RANGING_OFFSET_SET_TO           11954
508155.828 CMAC_LOG_RANGING_CONTINUE
508165.892 CMAC_LOG_RNG_REQ_TRANSMITTED
508165.894 CMAC_LOG_RNG_RSP_MSG_RCVD
508165.896 CMAC_LOG_ADJUST_TX_POWER                 -9
508165.898 CMAC_LOG_POWER_LEVEL_IS                  31.0  dBmV (commanded)
508165.900 CMAC_LOG_RANGING_CONTINUE
508175.962 CMAC_LOG_RNG_REQ_TRANSMITTED
508175.964 CMAC_LOG_RNG_RSP_MSG_RCVD
508175.966 CMAC_LOG_RANGING_SUCCESS
```

Event 5—Establish IP Connectivity

After ranging is complete, the cable interface on the cable modem is up. Now the cable modem accesses a remote DHCP server to get an IP address. The DHCP request also includes the name of a file that contains additional configuration parameters, the TFTP server's address, and the Time of Day (TOD) server's address.

The field CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS indicates the IP address assigned from the DHCP server to the cable modem interface. The field CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS marks the TFTP server's address. The field CMAC_LOG_DHCP_TOD_SERVER_ADDRESS indicates the time of day server's address. The field CMAC_LOG_DHCP_CONFIG_FILE_NAME shows the filename containing the transmission parameters. The field CMAC_LOG_DHCP_COMPLETE shows that all the IP connectivity was a success.

```
508175.968 CMAC_LOG_STATE_CHANGE                        dhcp_state
508176.982 CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS            188.188.1.62
508176.984 CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS            4.0.0.1
508176.986 CMAC_LOG_DHCP_TOD_SERVER_ADDRESS             4.0.0.32
508176.988 CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
508176.988 CMAC_LOG_DHCP_TZ_OFFSET                      360
508176.990 CMAC_LOG_DHCP_CONFIG_FILE_NAME               platinum.cm
508176.992 CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
508176.996 CMAC_LOG_DHCP_COMPLETE
```

Event 6—Establish the Time of Day

The Cisco uBR904 cable modem accesses the Time of Day server for the current date and time, which is used to create time stamps for logged events. The field CMAC_LOG_TOD_COMPLETE indicates a successful time of day sequence.

```
508177.120 CMAC_LOG_STATE_CHANGE                        establish_tod_state
508177.126 CMAC_LOG_TOD_REQUEST_SENT
508177.154 CMAC_LOG_TOD_REPLY_RECEIVED                  3107617539
508177.158 CMAC_LOG_TOD_COMPLETE
```

Event 7—Establish Security

The cable modem establishes a security association. The security_association_state is normally bypassed since "full security" as defined by MCNS DOCSIS is not supported.

"Full security" was a request made by MSOs for a very strong authorization and authentication check by the CMTS. This request was not granted by cable modem manufacturers. Cisco fully supports baseline privacy, which protects user's data from getting "sniffed" on the cable network.

```
508177.160 CMAC_LOG_STATE_CHANGE                        security_association_state
508177.162 CMAC_LOG_SECURITY_BYPASSED
```

Event 8—Transfer Operational Parameters

After the DHCP and security operations are successful, the cable modem downloads operational parameters from a cable company's TFTP server. These parameters are transferred via a configuration file. The field CMAC_LOG_DHCP_CONFIG_FILE_NAME shows the filename containing the transmission parameters.

```
508177.164 CMAC_LOG_STATE_CHANGE                      configuration_file_state
508177.166 CMAC_LOG_LOADING_CONFIG_FILE               platinum.cm
508178.280 CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
```

Event 9—Perform Registration

The cable modem registers with the headend Cisco uBR7246. After the cable modem is initialized, authenticated, and configured, the cable modem is authorized to forward traffic into the cable network. A successful registration is indicated by the field CMAC_LOG_REGISTRATION_OK.

```
508178.300 CMAC_LOG_STATE_CHANGE                      registration_state
508178.302 CMAC_LOG_REG_REQ_MSG_QUEUED
508178.306 CMAC_LOG_REG_REQ_TRANSMITTED
508178.310 CMAC_LOG_REG_RSP_MSG_RCVD
508178.312 CMAC_LOG_COS_ASSIGNED_SID                  1/2
508178.314 CMAC_LOG_RNG_REQ_QUEUED                    2
508178.316 CMAC_LOG_REGISTRATION_OK
```

Event 10—Comply with Baseline Privacy

Keys for baseline privacy are exchanged between the cable modem and the CMTS (the Cisco uBR7246 cable router). During this event, a link level encryption is performed so that a user's data cannot be "sniffed" by anyone else who is on the cable network.

Here is a trace that shows baseline privacy enabled. The key management protocol is responsible for exchanging 2 types of keys: KEKs and TEKs. The KEK (key exchange key, also referred to as the authorization key) is used by the headend CMTS to encrypt TEKs (traffic encryption keys) it sends to the cable modem. The TEKs are used to encrypt/decrypt the data. There is a TEK for each SID configured to use privacy.

```
   851.088 CMAC_LOG_STATE_CHANGE                      establish_privacy_state
   851.094 CMAC_LOG_PRIVACY_FSM_STATE_CHANGE          machine: KEK, event/state:
EVENT_1_PROVISIONED/STATE_A_START, new state: STATE_B_AUTH_WAIT
   851.102 CMAC_LOG_BPKM_REQ_TRANSMITTED
   851.116 CMAC_LOG_BPKM_RSP_MSG_RCVD
   851.120 CMAC_LOG_PRIVACY_FSM_STATE_CHANGE          machine: KEK, event/state:
EVENT_3_AUTH_REPLY/STATE_B_AUTH_WAIT, new state: STATE_C_AUTHORIZED
   856.208 CMAC_LOG_PRIVACY_FSM_STATE_CHANGE          machine: TEK, event/state:
EVENT_2_AUTHORIZED/STATE_A_START, new state: STATE_B_OP_WAIT
   856.220 CMAC_LOG_BPKM_REQ_TRANSMITTED
   856.224 CMAC_LOG_BPKM_RSP_MSG_RCVD
   856.230 CMAC_LOG_PRIVACY_FSM_STATE_CHANGE          machine: TEK, event/state:
EVENT_8_KEY_REPLY/STATE_B_OP_WAIT, new state: STATE_D_OPERATIONAL
   856.326 CMAC_LOG_PRIVACY_INSTALLED_KEY_FOR_SID     2
   856.330 CMAC_LOG_PRIVACY_ESTABLISHED
```

Event 11—Enter the Maintenance State

As soon as the cable modem is completely up and running, it enters the operational maintenance state.

```
508178.322 CMAC_LOG_STATE_CHANGE                    maintenance_state
```

Use Additional Troubleshooting Commands

You can use other show controller and debug cable modem commands to troubleshoot different aspects of a cable modem. However, the most useful command is the **show controller cable-modem 0 mac** command.

To display additional controller information inside a cable modem, enter one or more of the following commands in Privileged EXEC mode:

Command	Purpose
show controller cable-modem	Displays high-level controller information.
show controller cable-modem bpkm	Displays privacy state information.
show controller cable-modem des	Displays information about the Data Encryption Standard (DES) engine registers.
show controller cable-modem filters	Displays information about the MAC and SID cable modem filters.
show controller cable-modem lookup-table	Displays the mini-slot lookup table inside a cable modem.
show controller cable-modem mac [**errors** | **hardware** | **log** | **resets** | **state**]	Displays detailed MAC layer information.
show controller cable-modem phy	Displays physical layer information, such as receive and transmit physical registers.
show controller cable-modem tuner	Displays tuning information.

To debug different components of a cable modem, enter one or more of the following commands in Privileged EXEC mode:

Command	Purpose
debug cable-modem bpkm {**errors** | **events** | **packets**}	Debugs baseline privacy information.
debug cable-modem bridge	Debugs the bridge filter.
debug cable-modem error	Debugs cable interface errors.
debug cable-modem interrupts	Debugs cable modem interface interrupts.
debug cable-modem mac {**log** [*verbose*] | **messages**}	Displays and debugs the MAC layer log entries in real time.
debug cable-modem map	Debugs map message processing information.

Command Reference

This section provides new commands for troubleshooting the Cisco uBR904 cable modem.

All other commands used with this feature are documented in the Cisco IOS Release 11.3 command references.

- **show controller cable-modem**
- **show controller cable-modem bpkm**
- **show controller cable-modem mac**
- **show controller cable-modem phy**
- **show controller cable-modem des**
- **show controller cable-modem filters**
- **show controller cable-modem lookup-table**
- **show controller cable-modem tuner**
- **show interface cable-modem**

show controller cable-modem

To display high-level controller information about a cable modem, use the **show controller cable-modem** Privileged EXEC command.

Syntax Description

number	Controller number inside the cable modem.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

Following is a sample output for this command:

```
uBR904# show controller cable-modem 0
BCM Cable interface 0:
CM unit 0, idb 0x200EB4, ds 0x82D4748, regaddr = 0x800000, reset_mask 0x80
station address 0010.7b43.aa01  default station address 0010.7b43.aa01
PLD VERSION: 32

MAC State is ranging_2_state, Prev States = 7
MAC mcfilter 01E02F00  data mcfilter 01000000

DS: BCM 3116 Receiver: Chip id = 2
US: BCM 3037 Transmitter: Chip id = 30B4

Tuner: status=0x00
Rx: tuner_freq 699000000, symbol_rate 5055849, local_freq 11520000
    snr_estimate 33406, ber_estimate 0, lock_threshold 26000
    QAM in lock, FEC in lock, qam_mode QAM_64
Tx: tx_freq 20000000, power_level 0x3E, symbol_rate 1280000

DHCP: TFTP server = 4.0.0.32, TOD server = 4.0.0.188
      Security server = 0.0.0.0, Timezone Offest = 0.0.4.32
      Config filename =

buffer size 1600

RX data PDU ring with 32 entries at 0x201D40
   rx_head = 0x201D40 (0), rx_p = 0x82D4760 (0)

RX MAC message ring with 8 entries at 0x201E80
  rx_head_mac = 0x201EB8 (7), rx_p_mac = 0x82D4810 (7)

TX BD ring with 8 entries at 0x201FB8, tx_count = 0
  tx_head = 0x201FB8 (0), head_txp = 0x82D4888 (0)
  tx_tail = 0x201FB8 (0), tail_txp = 0x82D4888 (0)

TX PD ring with 8 entries at 0x202038, tx_count = 0
  tx_head_pd = 0x202038 (0)
  tx_tail_pd = 0x202038 (0)
```

```
Global control and status:
  global_ctrl_status=0x00
interrupts:
  irq_pend=0x0008, irq_mask=0x00F7
```

Related Commands

show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem lookup-table
show controller cable-modem mac
show controller cable-modem phy
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem bpkm

To display information about the baseline privacy key management exchange between the cable modem and the headend CMTS, use the **show controller cable-modem bpkm** Privileged EXEC command.

show controller cable-modem *number* **bpkm**

Syntax Description

number	Controller number inside the cable modem.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

The following output is displayed when the headend CMTS does not have baseline privacy enabled:

```
uBR904# show controller cable-modem 0 bpkm
CM Baseline Privacy Key Management
  configuration (in seconds):
      authorization wait time:    10
      reauthorization wait time:  10
      authorization grace time:   600
      operational wait time:      1
      rekey wait time:            1
      tek grace time:             600
      authorization rej wait time: 60
  kek state:    STATE_B_AUTH_WAIT
  sid 4:
      tek state: No resources assigned
```

Related Commands

show controller cable-modem
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem lookup-table
show controller cable-modem mac
show controller cable-modem phy
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem des

To display information about the Data Encryption Standard (DES) engine registers, use the **show controller cable-modem des** Privileged EXEC command.

show controller cable-modem des

Syntax Description

This command has no key words or arguments.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

DES engine registers are displayed in the following example:

```
uBR904# show controller cable-modem 0 des
downstream des:
  ds_des_key_table:
    key 0: even 0, odd 0
    key 1: even 0, odd 0
    key 2: even 0, odd 0
    key 3: even 0, odd 0
  ds_des_cbc_iv_table:
    iv  0: even 0, odd 0
    iv  1: even 0, odd 0
    iv  2: even 0, odd 0
    iv  3: even 0, odd 0
  ds_des_sid_table:
    sid_1=0x0000, sid_2=0x0000, sid_3=0x0000, sid_4=0x0000
  ds_des_sid_enable=0x80, ds_des_ctrl=0x2E
  ds_des_sv=0x0F00
  ds_unencrypted_length=0x0C
upstream des:
  us_des_key_table:
    key 0: even 0, odd 0
    key 1: even 0, odd 0
    key 2: even 0, odd 0
    key 3: even 0, odd 0
  us_des_cbc_iv_table:
    iv  0: even 0, odd 0
    iv  1: even 0, odd 0
    iv  2: even 0, odd 0
    iv  3: even 0, odd 0
  pb_req_bytes_to_minislots=0x10
  us_des_ctrl=0x00, us_des_sid_1= 0x1234
  ds_unencrypted_length=0x0C
```

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem filters
show controller cable-modem lookup-table
show controller cable-modem mac
show controller cable-modem phy
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem filters

To display the registers in the MAC hardware that are used for filtering received frames, use the **show controller cable-modem filters** Privileged EXEC command.

show controller cable-modem filters

Syntax Description

There are no key words or arguments for this command.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Some of the filtering parameters are MAC hardware addresses, Station IDs (SID), and upstream channel IDs.

This command is only useful for development engineers.

Sample Display

MAC and SID filter information is displayed in the following example:

```
uBR904# show controller cable-modem 0 filters
downstream mac message processing:
  ds_mac_da_filters:
    filter_1=0010.7b43.aa01, filter_2=0000.0000.0000
    filter_3=0000.0000.0000, filter_4=0000.0000.0000
  ds_mac_da_filter_ctrl=0x71, ds_mac_msg_sof=0x0000
  ds_mac_da_mc=01E02F00
  map_parser_sids:
    sid_1=0x0000, sid_2=0x0000, sid_3=0x0000, sid_4=0x0000
  ds_mac_filter_ctrl=0x00, us_channel_id=0x0000
  ds_pid=0x0000, mac_msg_proto_ver=FF 00
  reg_rang_req_sid=0x0000
downstream data processing:
  ds_data_da_filter_table:
    filter_1 0010.7b43.aa01, filter_2 0000.0000.0000
    filter_3 0000.0000.0000, filter_4 0000.0000.0000
  ds_data_da_filter_ctrl=0x61, ds_pdu_sof=0xDEAD
  ds_data_da_mc=01000000
upstream processing:
  us_ctrl_status=0x04, Minislots per request=0x01
  burst_maps:
    map[0]=0 map[1]=0 map[2]=0 map[3]=0
  bytes_per_minislot_exp=0x04
  us_map_parser_minislot_adv=0x03, maint_xmit=0x0000
  us_sid_table:
    sid_1=0x0000, sid_2=0x0000, sid_3=0x0000, sid_4=0x0000
  max_re_req=0x0010, rang_fifo=0x00
```

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem lookup-table
show controller cable-modem mac
show controller cable-modem phy
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem lookup-table

To display the mini-slot lookup table inside a cable modem, use the **show controller cable-modem lookup-table** Privileged EXEC command.

show controller cable-modem lookup-table

Syntax Description

This command has no keywords or arguments.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

This command shows the details of the lookup table. The driver uses this table to convert the size of a frame that the cable modem wants to transmit into a bandwidth request to the CMTS in "mini-slots." The contents of this table are affected by the upstream symbol rate that is negotiated between the CMTS and the cable modem.

This command is only useful for development engineers.

Sample Display

The lookup table is displayed in the following example:

```
uBR904# show controller cable-modem 0 lookup-table
PHY Overhead Lookup Table:

    01 01 01 01 01 01 01 02 02 02 02 02 02 02 02 02
    02 02 02 02 02 02 02 03 03 03 03 03 03 03 03 03
    03 03 03 03 03 03 03 04 04 04 04 04 04 04 04 04
    04 04 04 04 04 04 04 05 05 05 05 05 05 05 05 05
    05 05 05 05 05 05 05 06 06 06 06 06 06 06 06 06
    06 06 06 06 06 06 06 07 07 07 07 07 07 07 07 07
    07 07 07 07 07 07 07 08 08 08 08 08 08 08 08 08
    08 08 08 08 08 08 08 09 09 09 09 09 09 09 09 09
    09 09 09 09 09 09 09 0A 0A 0A 0A 0A 0A 0A 0A 0A
    0A 0A 0A 0A 0A 0A 0A 0B 0B 0B 0B 0B 0B 0B 0B 0B
    0B 0B 0B 0B 0B 0B 0B 0C 0C 0C 0C 0C 0C 0C 0C 0C
    0C 0C 0C 0C 0C 0C 0C 0D 0D 0D 0D 0D 0D 0D 0D 0D
    0D 0D 0D 0D 0D 0D 0D 0E 0E 0E 0E 0E 0E 0E 0E 0E
    0E 0E 0E 0E 0E 0E 0E 0F 0F 0F 0F 0F 0F 0F 0F 0F
    0F 0F 0F 0F 0F 0F 0F 10 10 10 10 10 10 10 10 10
    10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11
    11 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12
    12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13
    13 13 13 13 13 13 13 14 14 14 14 14 14 14 14 14
    14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15
    15 15 15 15 15 15 15 16 16 16 16 16 16 16 16 16
    16 16 16 16 16 16 16 17 17 17 17 17 17 17 17 17
    17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18
    18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19
    19 19 19 19 19 19 19 1A 1A 1A 1A 1A 1A 1A 1A 1A
    1A 1A 1A 1A 1A 1A 1A 1B 1B 1B 1B 1B 1B 1B 1B 1B
    1B 1B 1B 1B 1B 1B 1B 1C 1C 1C 1C 1C 1C 1C 1C 1C
```

```
1C 1C 1C 1C 1C 1C 1C 1D 1D 1D 1D 1D 1D 1D 1D 1D
1D 1D 1D 1D 1D 1D 1D 1E 1E 1E 1E 1E 1E 1E 1E 1E
1E 1E 1E 1E 1E 1E 1E 1F 1F 1F 1F 1F 1F 1F 1F 1F
1F 1F 1F 1F 1F 1F 1F 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22 22 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 25 25 25 25 25 25 25 25 25
25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26
26 26 26 26 26 26 26 27 27 27 27 27 27 27 27 27
27 27 27 27 27 27 27 28 28 28 28 28 28 28 28 28
28 28 28 28 28 28 28 29 29 29 29 29 29
29 29 29 29 29 29 29 2A 2A 2A 2A 2A 2A 2A 2A 2A
2A 2A 2A 2A 2A 2A 2A 2B 2B 2B 2B 2B 2B 2B 2B 2B
2B 2B 2B 2B 2B 2B 2B 2C 2C 2C 2C 2C 2C 2C 2C 2C
2C 2C 2C 2C 2C 2C 2C 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2E 2E 2E 2E 2E 2E 2E 2E 2E
2E 2E 2E 2E 2E 2E 2E 2F 2F 2F 2F 2F 2F 2F 2F 2F
2F 2F 2F 2F 2F 2F 2F 30 30 30 30 30 30 30 30 30
30 30 30 30 30 30 30 31 31 31 31 31 31 31 31 31
31 31 31 31 31 31 31 32 32 32 32 32 32 32 32 32
32 32 32 32 32 32 32 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 34 34 34 34 34 34 34 34 34
34 34 34 34 34 34 34 35 35 35 35 35 35 35 35 35
35 35 35 35 35 35 35 36 36 36 36 36 36 36 36 36
36 36 36 36 36 36 36 37 37 37 37 37 37 37 37 37
37 37 37 37 37 37 37 38 38 38 38 38 38 38 38 38
38 38 38 38 38 38 38 39 39 39 39 39 39 39 39 39
.
.
.
```

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem mac
show controller cable-modem phy
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem mac

To show detailed MAC layer information for a cable modem, enter the **show cable controller cable-modem** *number* **mac** Privileged EXEC command.

show controller cable-modem *number* **mac** [**errors | hardware | log | resets | state**]

Syntax Description

number	The controller number inside the cable modem.
errors	(Optional) Displays a log of the error events that are reported to SNMP. This keyword gives you a way of looking at the error events without using a MIB.
hardware	(Optional) Displays all MAC hardware registers.
log	(Optional) Displays a history of MAC log messages, up to 1023 entries. This is the same output that is displayed when the **debug cable mac log** command is entered.
resets	(Optional) Extracts all the reset causes out of the MAC log file and summarizes them into a mini report.
state	(Optional) Displays a summary of the MAC state.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

MAC log messages are written to a circular log file even when debugging is not turned on. These messages include timestamps, events, and information pertinent to these events. Enter the **show controller cable-modem** *number* **mac log** command to view MAC log messages.

If the cable modem interface fails to come up or resets periodically, the MAC log will capture what happened. For example, if an address is not obtained from the DHCP server, an error is logged, initialization starts over, and the cable modem scans for a downstream frequency.

The most useful keywords for troubleshooting a cable modem are **log**, **errors**, and **resets**. See Sample Display 1, Sample Display 2, and Sample Displays 3.

Sample Display 1

The following sample display shows the MAC log file for a cable-modem interface that has successfully come up:

```
uBR904# show controller cable-modem 0 mac log
*Mar  7 01:42:59: 528302.040 CMAC_LOG_LINK_DOWN
*Mar  7 01:42:59: 528302.042 CMAC_LOG_RESET_FROM_DRIVER
*Mar  7 01:42:59: 528302.044 CMAC_LOG_STATE_CHANGE                      wait_for_link_up_state
*Mar  7 01:42:59: 528302.046 CMAC_LOG_DRIVER_INIT_IDB_SHUTDOWN          0x08098D02
*Mar  7 01:42:59: 528302.048 CMAC_LOG_LINK_DOWN
*Mar  7 01:43:05: 528308.428 CMAC_LOG_DRIVER_INIT_IDB_RESET             0x08098E5E
*Mar  7 01:43:05: 528308.432 CMAC_LOG_LINK_DOWN
*Mar  7 01:43:05: 528308.434 CMAC_LOG_LINK_UP
*Mar  7 01:43:05: 528308.436 CMAC_LOG_STATE_CHANGE                      ds_channel_scanning_state
*Mar  7 01:43:05: 528308.440 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     88/453000000/855000000/6000000
*Mar  7 01:43:05: 528308.444 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     89/93000000/105000000/6000000
*Mar  7 01:43:05: 528308.448 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     90/111250000/117250000/6000000
*Mar  7 01:43:05: 528308.452 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     91/231012500/327012500/6000000
*Mar  7 01:43:05: 528308.456 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     92/333015000/333015000/6000000
*Mar  7 01:43:05: 528308.460 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     93/339012500/399012500/6000000
*Mar  7 01:43:05: 528308.462 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     94/405000000/447000000/6000000
*Mar  7 01:43:05: 528308.466 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     95/123015000/129015000/6000000
*Mar  7 01:43:05: 528308.470 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     96/135012500/135012500/6000000
*Mar  7 01:43:05: 528308.474 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     97/141000000/171000000/6000000
*Mar  7 01:43:05: 528308.478 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     98/219000000/225000000/6000000
*Mar  7 01:43:05: 528308.482 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND     99/177000000/213000000/6000000
*Mar  7 01:43:05: 528308.486 CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY    663000000
*Mar  7 01:43:05: 528308.488 CMAC_LOG_WILL_SEARCH_USER_DS_FREQUENCY     663000000
*Mar  7 01:43:07: 528310.292 CMAC_LOG_DS_64QAM_LOCK_ACQUIRED            663000000
*Mar  7 01:43:07: 528310.294 CMAC_LOG_DS_CHANNEL_SCAN_COMPLETED
*Mar  7 01:43:07: 528310.296 CMAC_LOG_STATE_CHANGE                      wait_ucd_state
*Mar  7 01:43:08: 528310.892 CMAC_LOG_UCD_MSG_RCVD                      4
*Mar  7 01:43:08: 528310.896 CMAC_LOG_UCD_NEW_US_FREQUENCY              20000000
*Mar  7 01:43:08: 528310.898 CMAC_LOG_SLOT_SIZE_CHANGED                 8
*Mar  7 01:43:08: 528310.970 CMAC_LOG_FOUND_US_CHANNEL                  1
*Mar  7 01:43:08: 528310.974 CMAC_LOG_STATE_CHANGE                      wait_map_state
*Mar  7 01:43:08: 528311.394 CMAC_LOG_MAP_MSG_RCVD
*Mar  7 01:43:08: 528311.396 CMAC_LOG_INITIAL_RANGING_MINISLOTS         40
*Mar  7 01:43:08: 528311.400 CMAC_LOG_STATE_CHANGE                      ranging_1_state
*Mar  7 01:43:08: 528311.402 CMAC_LOG_RANGING_OFFSET_SET_TO             9610
*Mar  7 01:43:08: 528311.404 CMAC_LOG_POWER_LEVEL_IS                    8.0  dBmV (commanded)
*Mar  7 01:43:08: 528311.406 CMAC_LOG_STARTING_RANGING
*Mar  7 01:43:08: 528311.408 CMAC_LOG_RANGING_BACKOFF_SET               0
*Mar  7 01:43:08: 528311.412 CMAC_LOG_RNG_REQ_QUEUED                    0
*Mar  7 01:43:09: 528311.900 CMAC_LOG_RNG_REQ_TRANSMITTED
*Mar  7 01:43:09: 528312.102 CMAC_LOG_T3_TIMER
*Mar  7 01:43:12: 528314.622 CMAC_LOG_POWER_LEVEL_IS                    20.0  dBmV (commanded)
*Mar  7 01:43:12: 528314.624 CMAC_LOG_RANGING_BACKOFF_SET               2
*Mar  7 01:43:12: 528314.628 CMAC_LOG_RNG_REQ_QUEUED                    0
*Mar  7 01:43:13: 528315.928 CMAC_LOG_RNG_REQ_TRANSMITTED
*Mar  7 01:43:13: 528315.932 CMAC_LOG_RNG_RSP_MSG_RCVD
*Mar  7 01:43:13: 528315.934 CMAC_LOG_RNG_RSP_SID_ASSIGNED              4
*Mar  7 01:43:13: 528315.936 CMAC_LOG_ADJUST_RANGING_OFFSET             2849
*Mar  7 01:43:13: 528315.938 CMAC_LOG_RANGING_OFFSET_SET_TO             12459
*Mar  7 01:43:13: 528315.940 CMAC_LOG_ADJUST_TX_POWER                   20
*Mar  7 01:43:13: 528315.942 CMAC_LOG_POWER_LEVEL_IS                    25.0  dBmV (commanded)
*Mar  7 01:43:13: 528315.944 CMAC_LOG_STATE_CHANGE                      ranging_2_state
*Mar  7 01:43:13: 528315.948 CMAC_LOG_RNG_REQ_QUEUED                    4
*Mar  7 01:43:14: 528316.942 CMAC_LOG_RNG_REQ_TRANSMITTED
*Mar  7 01:43:14: 528316.944 CMAC_LOG_RNG_RSP_MSG_RCVD
*Mar  7 01:43:14: 528316.946 CMAC_LOG_ADJUST_TX_POWER                   20
*Mar  7 01:43:14: 528316.950 CMAC_LOG_POWER_LEVEL_IS                    30.0  dBmV (commanded)
*Mar  7 01:43:14: 528316.952 CMAC_LOG_RANGING_CONTINUE
*Mar  7 01:43:15: 528317.956 CMAC_LOG_RNG_REQ_TRANSMITTED
```

```
*Mar  7 01:43:15: 528317.958 CMAC_LOG_RNG_RSP_MSG_RCVD
*Mar  7 01:43:15: 528317.960 CMAC_LOG_ADJUST_TX_POWER                    14
*Mar  7 01:43:15: 528317.962 CMAC_LOG_POWER_LEVEL_IS                     34.0  dBmV (commanded)
*Mar  7 01:43:15: 528317.964 CMAC_LOG_RANGING_CONTINUE
*Mar  7 01:43:16: 528318.968 CMAC_LOG_RNG_REQ_TRANSMITTED
*Mar  7 01:43:16: 528318.970 CMAC_LOG_RNG_RSP_MSG_RCVD
*Mar  7 01:43:16: 528318.974 CMAC_LOG_RANGING_SUCCESS
*Mar  7 01:43:16: 528318.976 CMAC_LOG_STATE_CHANGE                       dhcp_state
*Mar  7 01:43:16: 528318.978 DHCP_COMPLETE
*Mar  7 01:43:16: 528318.980 CMAC_LOG_STATE_CHANGE                       establish_tod_state
*Mar  7 01:43:16: 528318.982 CMAC_LOG_TOD_COMPLETE
*Mar  7 01:43:16: 528318.984 CMAC_LOG_STATE_CHANGE                       security_association_state
*Mar  7 01:43:16: 528318.986 CMAC_LOG_SECURITY_BYPASSED
*Mar  7 01:43:16: 528318.988 CMAC_LOG_STATE_CHANGE                       configuration_file_state
*Mar  7 01:43:16: 528318.992 CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
*Mar  7 01:43:16: 528319.028 CMAC_LOG_STATE_CHANGE                       registration_state
*Mar  7 01:43:16: 528319.030 CMAC_LOG_REG_REQ_MSG_QUEUED
*Mar  7 01:43:16: 528319.036 CMAC_LOG_REG_REQ_TRANSMITTED
*Mar  7 01:43:16: 528319.038 CMAC_LOG_REG_RSP_MSG_RCVD
*Mar  7 01:43:16: 528319.040 CMAC_LOG_COS_ASSIGNED_SID                   1/4
*Mar  7 01:43:16: 528319.044 CMAC_LOG_RNG_REQ_QUEUED                     4
*Mar  7 01:43:16: 528319.046 CMAC_LOG_REGISTRATION_OK
*Mar  7 01:43:16: 528319.048 CMAC_LOG_STATE_CHANGE                       establish_privacy_state
*Mar  7 01:43:16: 528319.052 CMAC_LOG_STATE_CHANGE                       maintenance_state
```

If the DHCP server could not be reached, the error would look like this in the MAC log:

```
497959.800 CMAC_LOG_STATE_CHANGE                       dhcp_state
497969.864 CMAC_LOG_RNG_REQ_TRANSMITTED
497969.866 CMAC_LOG_RNG_RSP_MSG_RCVD
497979.936 CMAC_LOG_RNG_REQ_TRANSMITTED
497979.938 CMAC_LOG_RNG_RSP_MSG_RCVD
497989.802 CMAC_LOG_WATCHDOG_TIMER
497989.804 CMAC_LOG_RESET_DHCP_WATCHDOG_EXPIRED
497989.804 CMAC_LOG_STATE_CHANGE                       reset_interface_state
497989.806 CMAC_LOG_DHCP_PROCESS_KILLED
```

Sample Display 2

MAC error log information is displayed in the following example, which is also reported via SNMP:

```
uBR904# show controller cable-modem 0 mac errors

 62856.934 R02.0  No Ranging Response received.  T3 time-out.
 62857.436 R02.0  No Ranging Response received.  T3 time-out.
 62859.450 R02.0  No Ranging Response received.  T3 time-out.
 62860.962 R02.0  No Ranging Response received.  T3 time-out.
 62908.796 D05.0  TFTP Request sent.  No Response/No Server.
 62949.080 D05.0  TFTP Request sent.  No Response/No Server.
 62989.368 D05.0  TFTP Request sent.  No Response/No Server.
 63029.650 D05.0  TFTP Request sent.  No Response/No Server.
 63069.932 D05.0  TFTP Request sent.  No Response/No Server.
```

If the DHCP server could not be reached, the error would look like this in the MAC error display:

```
uBR904# show controller cable-modem 0 mac errors

497989.804 D01.0  Discover sent no Offer received.  No available DHCP Server.
498024.046 D01.0  Discover sent no Offer received.  No available DHCP Server.
498058.284 D01.0  Discover sent no Offer received.  No available DHCP Server.
```

Sample Displays 3

The **show controller cable-modem 0 mac resets** command shows only the entries in the cable MAC log that begin with the field CMAC_LOG_RESET. Collectively presenting these fields provides you with a summary of the most recent reasons why the cable interface was reset.

Reset messages and brief explainations are included in the following examples and in Table 1. However, the reset messages in Table 2 do not commonly occur.

In the following example, the configuration file downloaded from the TFTP server could not be read. The file might not exist, or the file has incorrect permissions.

```
uBR904# show controller cable-modem 0 mac resets

 62526.114 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 62564.368 CMAC_LOG_RESET_T4_EXPIRED
 62677.178 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 62717.462 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 62757.746 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 62796.000 CMAC_LOG_RESET_T4_EXPIRED
 62908.808 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 62949.092 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 62989.380 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 63029.662 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 63069.944 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 63110.228 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
 63148.484 CMAC_LOG_RESET_T4_EXPIRED
 63261.296 CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
```

The following example shows that the DHCP server could not be reached. The DHCP server took too long to respond.

```
uBR904# show controller cable-modem 0 mac resets

497989.804 CMAC_LOG_RESET_DHCP_WATCHDOG_EXPIRED
498024.046 CMAC_LOG_RESET_DHCP_WATCHDOG_EXPIRED
498058.284 CMAC_LOG_RESET_DHCP_WATCHDOG_EXPIRED
```

This next example indicates that an event in the cable interface driver caused the interface to reset. This is often because a shut or clear command is currently being issued on the interface.

```
uBR904# show controller cable-modem 0 mac resets
527986.444 CMAC_LOG_RESET_FROM_DRIVER
528302.042 CMAC_LOG_RESET_FROM_DRIVER
528346.600 CMAC_LOG_RESET_FROM_DRIVER
528444.494 CMAC_LOG_RESET_FROM_DRIVER
```

Table 2 Possible but Uncommon Cable Interface Reset Causes

Message	Description
CMAC_LOG_RESET_CONFIG_FILE_PARSE_FAILED	The format of the DOCSIS configuration file acquired from the TFTP server is not acceptable.
CMAC_LOG_RESET_LOSS_OF_SYNC	Synchronization with the CMTS has been lost (SYNC messages are not being received).
CMAC_LOG_RESET_T4_EXPIRED	Maintenance ranging opportunities for this modem are not being received from the CMTS.
CMAC_LOG_RESET_DHCP_WATCHDOG_EXPIRED	The DHCP server took too long to respond.
CMAC_LOG_RESET_TOD_WATCHDOG_EXPIRED	The Time Of Day server took too long to respond.

Table 2 Possible but Uncommon Cable Interface Reset Causes (Continued)

Message	Description
CMAC_LOG_RESET_PRIVACY_WATCHDOG_EXPIRED	The baseline privacy exchange with the CMTS took too long.
CMAC_LOG_RESET_CHANGE_US_WATCHDOG_EXPIRED	The cable modem was unable to transmit a response to a UCC-REQ message.
CMAC_LOG_RESET_SECURITY_WATCHDOG_EXPIRED	The "full security" exchange with the CMTS took too long.
CMAC_LOG_RESET_CONFIG_FILE_WATCHDOG_EXPIRED	The TFTP server took too long to respond.
CMAC_LOG_RESET_ALL_FREQUENCIES_SEARCHED	All downstream frequencies to be searched have been searched. This message indicates that downstream frequencies were found and the cable modem failed.
CMAC_LOG_RESET_T2_EXPIRED	Initial ranging opportunities are not being received.
CMAC_LOG_RESET_T3_RETRIES_EXHAUSTED	The CMTS failed to respond to a RNG-REQ message too many times.
CMAC_LOG_RESET_RANGING_ABORTED	The CMTS commanded the cable modem to abort the ranging process.
CMAC_LOG_RESET_NO_MEMORY	The cable modem has run out of memory.
CMAC_LOG_RESET_CANT_START_PROCESS	The cable modem was unable to start an internal process necessary to complete ranging and registration.
CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED	The reading of the configuration file from the TFTP server failed. The file might not exist, or it has incorrect permissions.
CMAC_LOG_RESET_AUTHENTICATION_FAILURE	The cable modem failed authentication, as indicated in a REG-RSP message from the CMTS.
CMAC_LOG_RESET_SERVICE_NOT_AVAILABLE	The CMTS has failed the cable modem's registration because a required or requested Class Of Service is not available.
CMAC_LOG_RESET_T6_RETRIES_EXHAUSTED	The CMTS failed to respond to a REG-REQ message too many times.
CMAC_LOG_RESET_MAINTENANCE_WATCHDOG_DRIVER	The cable modem MAC layer failed to detect a change in the interface driver.
CMAC_LOG_RESET_NET_ACCESS_MISSING	The Network Access parameter was missing from the DOCSIS configuration file.
CMAC_LOG_RESET_FAILED_WRITE_ACCESS_CONTROL	The cable modem was unable to set the Write Access Control for an SNMP parameter, as specified by the DOCSIS configuration file.
CMAC_LOG_RESET_DHCP_FAILED	The DHCP server did not respond with all the required values. The required values are: IP address, network mask, TFTP server IP address, time server IP address, DOCSIS configuration file name, and time zone offset.
CMAC_LOG_RESET_CANT_START_DS_TUNER_PRCESS	The modem was unable to start the internal process used to manage the downstream tuner.
CMAC_LOG_RESET_TOO_MANY_DS_LOCKS_LOST	Downstream QAM/FEC lock has been lost too many times.

Table 2 Possible but Uncommon Cable Interface Reset Causes (Continued)

Message	Description
CMAC_LOG_RESET_NO_SEND_TO_DS_TUNER_PROCESS	The modem MAC layer process was unable to communicate with the downstream tuner management process.
CMAC_LOG_RESET_DS_TUNER_WATCHDOG	The downstream tuner process failed to report it's continuing operation for a long period of time.
CMAC_LOG_RESET_UNABLE_TO_SET_MIB_OBJECT	The cable modem was unable to set an SNMP parameter as specified by the DOCSIS configuration file.
CMAC_LOG_RESET_MIB_OBJECT_PROCESS_WATCHDOG	The internal MIB object took to long to process the entries in the DOCSIS configuration file.

Sample Display 4

This example display for the **show controller cable-modem 0 mac hardware** command shows the detailed configuration of the interface driver and MAC layer hardware. The most interesting bit is the station address (hardware address). The rest of the display is only of use to a software engineer. The MIB statistics reflect the MAC hardware counters for various events, but these counters are typically reset every few seconds, so their contents are not accurate in this display.

```
uBR904# show controller cable-modem 0 mac hardware
PLD VERSION: 32

BCM3220 unit 0, idb 0x200EB4, ds 0x82D4748, regaddr = 0x800000, reset_mask
0x80
station address 0010.7b43.aa01  default station address 0010.7b43.aa01
MAC mcfilter 01E02F00  data mcfilter 01000000

buffer size 1600
RX data PDU ring with 32 entries at 0x201D40
  rx_head = 0x201D40 (0), rx_p = 0x82D4760 (0)
    00 pak=0x82DF844 buf=0x227F1A status=0x80 pak_size=0
    01 pak=0x82E0BF4 buf=0x22C56A status=0x80 pak_size=0
    02 pak=0x82DF454 buf=0x22710A status=0x80 pak_size=0
    03 pak=0x82DF64C buf=0x227812 status=0x80 pak_size=0
    04 pak=0x82E0024 buf=0x229B3A status=0x80 pak_size=0
    05 pak=0x82DBF2C buf=0x21B332 status=0x80 pak_size=0
    06 pak=0x82DFE2C buf=0x229432 status=0x80 pak_size=0
    07 pak=0x82E0FE4 buf=0x22D37A status=0x80 pak_size=0
    08 pak=0x82DF064 buf=0x2262FA status=0x80 pak_size=0
    09 pak=0x82DEC74 buf=0x2254EA status=0x80 pak_size=0
    10 pak=0x82DEA7C buf=0x224DE2 status=0x80 pak_size=0
    11 pak=0x82DE884 buf=0x2246DA status=0x80 pak_size=0
    12 pak=0x82DE68C buf=0x223FD2 status=0x80 pak_size=0
    13 pak=0x82DE494 buf=0x2238CA status=0x80 pak_size=0
    14 pak=0x82DE29C buf=0x2231C2 status=0x80 pak_size=0
    15 pak=0x82DE0A4 buf=0x222ABA status=0x80 pak_size=0
    16 pak=0x82DDEAC buf=0x2223B2 status=0x80 pak_size=0
    17 pak=0x82DDCB4 buf=0x221CAA status=0x80 pak_size=0
    18 pak=0x82DDABC buf=0x2215A2 status=0x80 pak_size=0
    19 pak=0x82DD8C4 buf=0x220E9A status=0x80 pak_size=0
    20 pak=0x82DD6CC buf=0x220792 status=0x80 pak_size=0
    21 pak=0x82DD4D4 buf=0x22008A status=0x80 pak_size=0
    22 pak=0x82DD2DC buf=0x21F982 status=0x80 pak_size=0
    23 pak=0x82DD0E4 buf=0x21F27A status=0x80 pak_size=0
    24 pak=0x82DCEEC buf=0x21EB72 status=0x80 pak_size=0
    25 pak=0x82DCCF4 buf=0x21E46A status=0x80 pak_size=0
    26 pak=0x82DCAFC buf=0x21DD62 status=0x80 pak_size=0
```

```
    27 pak=0x82DC904 buf=0x21D65A status=0x80 pak_size=0
    28 pak=0x82DC70C buf=0x21CF52 status=0x80 pak_size=0
    29 pak=0x82DC514 buf=0x21C84A status=0x80 pak_size=0
    30 pak=0x82DC31C buf=0x21C142 status=0x80 pak_size=0
    31 pak=0x82DC124 buf=0x21BA3A status=0xA0 pak_size=0
RX MAC message ring with 8 entries at 0x201E80
  rx_head_mac = 0x201EB0 (6), rx_p_mac = 0x82D480C (6)
    00 pak=0x82E0DEC buf=0x22CC72 status=0x80 pak_size=0
    01 pak=0x82E021C buf=0x22A242 status=0x80 pak_size=0
    02 pak=0x82E060C buf=0x22B052 status=0x80 pak_size=0
    03 pak=0x82E11DC buf=0x22DA82 status=0x80 pak_size=0
    04 pak=0x82DFC34 buf=0x228D2A status=0x80 pak_size=0
    05 pak=0x82E09FC buf=0x22BE62 status=0x80 pak_size=0
    06 pak=0x82DEE6C buf=0x225BF2 status=0x80 pak_size=0
    07 pak=0x82DFA3C buf=0x228622 status=0xA0 pak_size=0
TX BD ring with 8 entries at 0x201FB8, tx_count = 0
  tx_head = 0x201FB8 (0), head_txp = 0x82D4888 (0)
  tx_tail = 0x201FB8 (0), tail_txp = 0x82D4888 (0)
    00 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    01 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    02 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    03 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    04 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    05 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    06 pak=0x000000 buf=0x200000 status=0x00 pak_size=0
    07 pak=0x000000 buf=0x200000 status=0x20 pak_size=0
TX PD ring with 8 entries at 0x202038, tx_count = 0
  tx_head_pd = 0x202038 (0)
  tx_tail_pd = 0x202038 (0)
    00 status=0x00 bd_index=0x0000 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 2E FF FF
    01 status=0x00 bd_index=0x0001 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 2E FF FF
    02 status=0x00 bd_index=0x0002 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 2E FF FF
    03 status=0x00 bd_index=0x0003 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 2E FF FF
    04 status=0x00 bd_index=0x0004 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 2E 00 00
    05 status=0x00 bd_index=0x0005 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 2E 00 00
    06 status=0x00 bd_index=0x0006 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 00 00 00
    07 status=0x20 bd_index=0x0007 len=0x0000 hdr_len=0x0000
    ehdr:  00 00 00 00 00 00

MIB Statistics
  DS fifo full = 0, Rerequests = 0
  DS mac msg overruns = 0, DS data overruns = 0
  Qualified maps = 0, Qualified syncs = 0
  CRC fails = 0, HDR chk fails = 0
  Data pdus = 0, Mac msgs = 0
  Valid hdrs = 0
BCM3220 Registers:
downstream dma:
  ds_data_bd_base=0x001D40, ds_mac_bd_base=0x001E80
  ds_data_dma_ctrl=0x98, ds_mac_dma_ctrl=0x98
  ds_dma_data_index=0x0000, ds_dma_msg_index=0x0000
upstream dma:
  us_bd_base=0x001FB8, us_pd_base=0x002038
  us_dma_ctrl=0x00, us_dma_tx_start=0x00
global control and status:
  global_ctrl_status=0x00
interrupts:
  irq_pend=0x0018, irq_mask=0x00E7
```

```
timing recovery circuit:
  loop_enable=0x00, minislot_divisor=0x00
  K0_ctrl=0x06, K1_ctrl=0x07, acq_threshhold=0x01
  err_threshhold=0x04, timeout_threshold=0xFF
  nco_bias=0x4F7004F7, ranging_offset=0x00000000
  ts_err=0x00, sync_valid=0x00, delta_F=0x00
  timeout_err=0x00
spi:
  dynamic_ctrl=0x09, static_ctr=0x9F, autonomous=0x01
  irq_ack=0x00, spi_cmd=0x51, spi_addr=0x11
  spi_data= FF/00/00/00/00/00/00
burst profiles:
  profile 0:
             01 19 1D 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  profile 1:
             01 19 1D 03 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  profile 2:
             01 19 1D 04 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  profile 3:
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
             00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Sample Display 5

The **show controller cable-modem 0 mac state** command summarizes the state of the cable MAC layer. If the cable MAC layer is in the `wait_for_link_up_state`, the information shown in the display corresponds to the last time the interface was up. This allows useful information to be acquired from this display even though the modem has not been able to range and register. The normal operational state of the interface is the `maintenance_state`.

```
uBR904# show controller cable-modem 0 mac state

MAC State:                  maintenance_state
Ranging SID:                5
Registered:                 TRUE
Privacy Established:        TRUE
DS ID:                      1
DS Frequency:               663000000
DS Symbol Rate:             5056941
DS QAM Mode                 64QAM
DS Search:
  88 453000000 855000000 6000000
  89  93000000 105000000 6000000
  90 111250000 117250000 6000000
  91 231012500 327012500 6000000
  92 333015000 333015000 6000000
  93 339012500 399012500 6000000
  94 405000000 447000000 6000000
  95 123015000 129015000 6000000
  96 135012500 135012500 6000000
  97 141000000 171000000 6000000
```

```
  98 219000000 225000000 6000000
  99 177000000 213000000 6000000
US ID:                        1
US Frequency:                 20000000
US Power Level:               34.0 (dBmV)
US Symbol Rate:               1280000
Ranging Offset:               12460
Mini-Slot Size:               8
Change Count:                 4
Preamble Pattern:             CC CC CC CC CC CC CC CC  CC CC CC CC CC CC 0D 0D
Burst Descriptor  0:
  Interval Usage Code:        1
  Modulation Type:            1
  Differential Encoding:      2
  Preamble Length:            64
  Preamble Value Offset:      56
  FEC Error Correction:       0
  FEC Codeword Info Bytes:    16
  Scrambler Seed:             338
  Maximum Burst Size:         1
  Guard Time Size:            8
  Last Codeword Length:       1
  Scrambler on/off:           1
Burst Descriptor  1:
  Interval Usage Code:        3
  Modulation Type:            1
  Differential Encoding:      2
  Preamble Length:            128
  Preamble Value Offset:      0
  FEC Error Correction:       5
  FEC Codeword Info Bytes:    34
  Scrambler Seed:             338
  Maximum Burst Size:         0
  Guard Time Size:            48
  Last Codeword Length:       1
  Scrambler on/off:           1
Burst Descriptor  2:
  Interval Usage Code:        4
  Modulation Type:            1
  Differential Encoding:      2
  Preamble Length:            128
  Preamble Value Offset:      0
  FEC Error Correction:       5
  FEC Codeword Info Bytes:    34
  Scrambler Seed:             338
  Maximum Burst Size:         0
  Guard Time Size:            48
  Last Codeword Length:       1
  Scrambler on/off:           1
Burst Descriptor  3:
  Interval Usage Code:        5
  Modulation Type:            1
  Differential Encoding:      2
  Preamble Length:            72
  Preamble Value Offset:      48
  FEC Error Correction:       5
  FEC Codeword Info Bytes:    75
  Scrambler Seed:             338
  Maximum Burst Size:         0
  Guard Time Size:            8
  Last Codeword Length:       1
  Scrambler on/off:           1
Config File:
Network Access:               TRUE
  Vendor ID:                  0.240.30
```

```
    Baseline Privacy:
    Auth. Wait Timeout:     10
    Reauth. Wait Timeout:   10
    Auth. Grace Time:       600
    Op. Wait Timeout:       1
    Retry Wait Timeout:     1
    TEK Grace Time:         600
    Auth. Reject Wait Time: 60
  COS  1:
    Assigned SID:           5
    Max Downstream Rate:    4000000
    Max Upstream Rate:      2000000
    Upstream Priority:      7
    Min Upstream Rate:      100000
    Max Upstream Burst:     12
    Privacy Enable:         TRUE
Ranging Backoff Start:      0 (at initial ranging)
Ranging Backoff End:        4 (at initial ranging)
Data Backoff Start:         0 (at initial ranging)
Data Backoff End:           4 (at initial ranging)
IP Address:                 0.0.0.0
Net Mask:                   0.0.0.0
TFTP Server IP Address:     223.255.254.254
Time Server IP Address:     188.188.1.5
Config File Name:           muck/ebuell/tftp/cm_conf
Time Zone Offset:           -28800
```

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem lookup-table
show controller cable-modem phy
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem phy

To display detailed contents about the registers used in the downstream physical hardware used by a cable modem, use the **show controller cable-modem phy** Privileged EXEC command.

show controller cable-modem phy {**receive** | **transmit**}

Syntax Description

receive	Displays all receive registers in the downstream physical hardware.
transmit	Displays all transmit registers in the upstream physical hardware.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

This command is only useful for development engineers.

Sample Display

Physical receive registers are displayed in the following example:

```
uBR904# show controller cable-modem 0 phy receive
BCM3116 Receiver Registers: Chip ID = C2C1

  rstctl=    frzctl=20  qamctl=1B  lmsctl=0B  tpctl=00  fmtctl=24
  ffectl=3F  irqsts=09  irqmask=00 stoscm=9E  rstctr=00  frzctl2=46
  dvctl=30   idepth=55  eqlctl=00  tstctl=02  berctl=00  clkset=00
  tunset=00  tunctl=03

FFC coefficient registers:
  F0=0067FFBC  F1=FF880080  F2=00C1FEFB  F3=FF75019D
  F4=00C5FD89  F5=FF6D0485  F6=FC95F690  F7=2D280000

DFE coefficient registers:
  D00=0636031E  D01=FBDD0314  D02=0077FD39  D03=001B00C6
  D04=0024FF74  D05=0015007E  D06=000CFFC4  D07=FFC0004B
  D08=0044FFF6  D09=FFE00019  D10=00190005  D11=FFD3FFAD
  D12=FFD3FFE0  D13=001A000A  D14=FFF3FFED  D15=0008FFFD
  D16=FFFC0024  D17=0023FFDF  D18=0029FFFF  D19=000D001E
  D20=00020017  D21=00250001  D22=0007FFF4  D23=FFF60014

ldsft=B0EE      ldsnre=0098AF  ldif=0D004E   ldbbi=00000000
ldbbq=00000000  ldali=032E00   ldaii=E62AF2  ldbrfo=705A05
ldbri=F9CDC200  lddrfo=007E7D  lddri=007EF0

FEC correctable error count:   0
FEC uncorrectable error count: 0
Bit Error Rate Count: 0
```

Physical transmit registers are displayed in the following example:

```
uBR904# show controller cable-modem 0 phy transmit
BCM3037 Transmitter Registers:
part_id     = 3037       rev_id      = 01
test_mode   = 00         test_input  = 00
test_misc   = 2009       rst         = 00
power       = 0000       power_2     = 00
port        = 6F         pll         = F7
map         = 66         mod         = 28
tx_oen_bdly = 14         tx_oen_edly = C8
prbs_cfg    = 00C000     baud        = 1A36E3
burst       = 0000       if_freq     = 200000
dac         = 37         tx_config   = 00

burst config 0 : prbs_init    = FFFFFF   rs      = 343E
                 fec          = 00       qam     = 01
                 pream_len    = 0018     offset  = 0000
burst config 1 : prbs_init    = FFFFFE   rs      = 033B
                 fec          = 1C       qam     = 65
                 pream_len    = 0000     offset  = 0000
burst config 2 : prbs_init    = FFFFFE   rs      = 033B
                 fec          = 1D       qam     = 65
                 pream_len    = 0000     offset  = 0000
burst config 3 : prbs_init    = FFFFFE   rs      = 033B
                 fec          = 1E       qam     = 65
burst config 4 : prbs_init    = FFFFFE   rs      = 033B
                 fec          = 1F       qam     = 65
                 pream_len    = 0000     offset  = 0000
burst config 5 : prbs_init    = FFFFFE   rs      = 033B
                 fec          = 0F       qam     = 66
                 pream_len    = 0000     offset  = 0000
Eq Coeff:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Preamble values:
CC CC CC CC CC 0D 0D CC CC CC CC CC CC CC CC 0D
04 25 01 01 01 01 02 01 02 03 02 00 40 04 02 00
40 05 01 00 06 01 10 07 02 01 52 08 01 01 09 01
08 0A 01 01 0B 01 02 04 25 03 01 01 01 02 01 02
03 02 00 50 04 02 00 30 05 01 00 06 01 22 07 02
01 52 08 01 00 09 01 30 0A 01 01 0B 01 02 04 25
04 01 01 01 02 01 02 03 02 00 40 04 02 00 40 05
01 00 06 01 22 07 02 01 52 08 01 00 09 01 30 0A
```

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem lookup-table
show controller cable-modem mac
show controller cable-modem tuner
show interface cable-modem

show controller cable-modem tuner

To display the settings for the upstream and downstream tuners used by a cable modem, use the **show controller cable-modem tuner** Privileged EXEC command.

show controller cable-modem tuner

Syntax Description

There are no key words or arguments for this command.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

The cable modem's tuner settings are displayed in the following example. See Table 3 for output field possibilities and descriptions.

```
uBR904# show controller cable-modem 0 tuner
Tuner: status=0x00
Rx: tuner_freq 507000000, symbol_rate 5360736, local_freq 11520000
    snr_estimate 17488, ber_estimate 0, lock_threshold 26000
    QAM not in lock, FEC not in lock, qam_mode QAM_64
Tx: tx_freq 20000000, power_level 0x3E, symbol_rate 1280000
```

Table 3 Show Controller Cable-Modem Tuner Field Descriptions

Field	Description
`tuner_freq`	Indicates the current downstream frequency.
`symbol_rate`	Indicates the downstream or upstream symbol rate in use.
`snr_estimate`	Signal to noise estimate in dB X 1000.
`ber_estimate`	Bit error rate estimate (always 0).
`QAM` *`status`*	Indicates if QAM/FEC loc has been acquired and the modulation mode in use.
`tx_freq`	Current upstream frequency.
`power_level`	Transmit power level as set in the hardware. The units are unique to the hardware used. Use the **show controller cable-modem 0 mac state** EXEC command to see the power level in dBmV.

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem lookup-table

show controller cable-modem mac
show controller cable-modem phy
show interface cable-modem

show interface cable-modem

To display information about the cable modem interface on a cable modem, use the **show interface cable-modem** EXEC command.

show interface cable-modem *number* [**accounting | counters | crb | irb | type**]

Syntax Description

number	Cable modem interface number.
accounting	(Optional) Displays the number of packets of each protocol type that has been sent through the cable modem interface.
counters	(Optional) Shows MIB counters on the cable interface.
crb	(Optional) Displays routing and bridging information pertaining to the cable interface.
irb	(Optional) Displays routing and bridging information pertaining to the cable interface.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

Traffic passing through the cable modem interface is shown in the following example:

```
uBR904# show interface cable-modem 0
cable-modem0 is up, line protocol is up
  Hardware is BCM3220, address is 0010.7b43.aa01 (bia 0010.7b43.aa01)
  Internet address is 188.188.1.60/16
  MTU 1500 bytes, BW 27000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation , loopback not set, keepalive not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:07:04, output 00:00:41, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4495 packets input, 1153221 bytes, 0 no buffer
     Received 8 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     12841 packets output, 1708272 bytes, 0 underruns
     0 output errors, 0 collisions, 11 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

The following example displays the number of packets and each protocol type passing through the cable modem interface:

```
uBR904# show int cable-modem 0 accounting
cable-modem0
                Protocol    Pkts In   Chars In   Pkts Out  Chars Out
                      IP        545     185502        159      90240
           Trans. Bridge       3878     964995      12597    1611142
                     ARP         73       3066         86       4128
```

MIB counters on the cable interface are displayed in the next example:

```
uBR904# show int cable-modem 0 counters
Cable specific counters:
Ranging requests sent  : 50982
Downstream FIFO full   : 0
Re-requests            : 7277
DS MAC Message Overruns: 0
DS Data Overruns       : 0
Received MAPs          : 254339485
Received Syncs         : 53059555
Message CRC failures   : 0
Header CRC failures    : 1394
Data PDUs              : 5853
DS MAC messages        : 307861745
Valid Headers          : 307869065
Sync losses            : 0
Pulse losses           : 1
BW request failures    : 6
```

Routing and bridging information on the cable modem interface is displayed in the next example:

```
uBR904# show int cable-modem 0 crb

cable-modem0

 Bridged protocols on cable-modem0:
  ip

 Software MAC address filter on cable-modem0
  Hash Len    Address      Matches  Act      Type
  0x00:  0 ffff.ffff.ffff      3877 RCV Physical broadcast
  0x2A:  0 0900.2b01.0001         0 RCV DEC spanning tree
  0x7A:  0 0010.7b43.aa01       573 RCV Interface MAC address
  0xC2:  0 0180.c200.0000         0 RCV IEEE spanning tree
  0xC2:  1 0180.c200.0000         0 RCV IBM spanning tree
soho5#show int cable-modem 0 type ?
  ethernet  Show ethernet vlan type
  trbrf     Show BRF tokenring vlan type
```

Related Commands

show controller cable-modem
show controller cable-modem bpkm
show controller cable-modem des
show controller cable-modem filters
show controller cable-modem lookup-table
show controller cable-modem mac
show controller cable-modem phy
show controller cable-modem tuner

Debug Commands

The following new debug commands are available to troubleshoot a cable modem:

- **debug cable-modem bpkm**
- **debug cable-modem bridge**
- **debug cable-modem error**
- **debug cable-modem interrupts**
- **debug cable-modem mac**
- **debug cable-modem map**

debug cable-modem bpkm

To debug baseline privacy information on a cable modem, use the **debug cable-modem mac** Privileged EXEC command. The **no** form of this command turns debugging messages off.

[**no**] **debug cable-modem bpkm** {**errors** | **events** | **packets**}

Syntax Description

errors	Debugs cable modem privacy errors.
events	Debugs events related to cable baseline privacy.
packets	Debugs baseline privacy packets.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

Figure 3 shows the required keywords within the **debug cable-modem bpkm** command. You must choose one.

Figure 3 Sample Debug Cable-Modem Bpkm Output

```
uBR904# debug cable-modem bpkm ?
  errors   Cable Modem privacy errors
  events   events related to cable baseline privacy
  packets  baseline privacy packets
```

Figure 4 shows output when the headend does not have privacy enabled.

Figure 4 Sample Debug Cable-Modem Bpkm Output

```
uBR904# debug cable bpkm
cm_bpkm_fsm(): machine: KEK, event/state: EVENT_4_TIMEOUT/STATE_B_AUTH_WAIT, new state:
STATE_B_AUTH_WAIT

cm_bpkm_fsm(): machine: KEK, event/state: EVENT_4_TIMEOUT/STATE_B_AUTH_WAIT, new state:
STATE_B_AUTH_WAIT

%LINEPROTO-5-UPDOWN: Line protocol on Interface cable-modem0, changed state to down
cm_bpkm_fsm(): machine: KEK, event/state: EVENT_1_PROVISIONED/STATE_A_START, new state:
STATE_B_AUTH_WAIT

%LINEPROTO-5-UPDOWN: Line protocol on Interface cable-modem0, changed state to up
```

Related Commands

debug cable-modem bridge
debug cable-modem error
debug cable-modem interrupts
debug cable-modem mac
debug cable-modem map

debug cable-modem bridge

Use the **debug cable-modem bridge** Privileged EXEC command to debug bridge filter processing information on a cable modem. The **no** form of this command turns debugging messages off.

[**no**] **debug cable-modem bridge**

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

When the interface is down, all bridge table entries learned on the Ethernet interface are set to discard because traffic is not bridged until the cable interface has completed initialization. After the interface is completely up (the line protocol), bridge table entries learned on the Ethernet interface program the cable's MAC data filters. The cable MAC hardware filters out any received packets whose addresses, are not in the filters. In this way, the cable interface only receives packets addressed to its own MAC address or an address it has learned on the Ethernet interface.

Sample Display

Figure 5 shows sample display output for the **debug cable-modem bridge** Privileged EXEC command

Figure 5 Sample Debug Cable-Modem Bridge Output

```
uBR904# debug cable-modem bridge
%LINEPROTO-5-UPDOWN: Line protocol on Interface cable-modem0, changed state to downshut
cm_tbridge_add_entry(): MAC not initialized, discarding entry: 00e0.fe7a.186fno shut
cm_tbridge_add_entry(): MAC not initialized, discarding entry: 00e0.fe7a.186f
%LINEPROTO-5-UPDOWN: Line protocol on Interface cable-modem0, changed state to up
cm_tbridge_add_entry(): Adding entry 00e0.fe7a.186f to filter 2
```

Related Commands

debug cable-modem bpkm
debug cable-modem error
debug cable-modem interrupts
debug cable-modem mac
debug cable-modem map

debug cable-modem error

Use the the **debug cable-modem error** Privileged EXEC command to enable debugging messages for the cable interface driver. The **no** form of this command turns debugging messages off.

[**no**] **debug cable-modem error**

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

This command displays detailed output about the sanity checking of received frame formats, the acquisition of downstream QAM/FEC lock, the receipt or non receipt of SYNC messages from the CMTS, reception errors, and bandwidth request failures.

Sample Display

Figure 6 shows sample display output for the **debug cable-modem error** command.

Figure 6 Sample Debug Cable-Modem Error Output

```
uBR904# debug cable-modem error
*Mar  7 20:16:29: AcquireSync(): Update rate is 100 Hz
*Mar  7 20:16:30: 1st Sync acquired after 1100 ms.
*Mar  7 20:16:30: Recovery loop is locked (7/9)
*Mar  7 20:16:30: 2nd Sync acquired after 100 ms.
*Mar  7 20:16:30: Recovery loop is locked (10/15)
```

Related Commands

debug cable-modem bpkm
debug cable-modem bridge
debug cable-modem interrupts
debug cable-modem mac
debug cable-modem map

debug cable-modem interrupts

Use the **debug cable-modem interrupts** command to debug cable modem interrupts. The **no** form of this command turns debugging messages off.

[**no**] **debug cable-modem interrupts**

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

Figure 7 shows sample debug output for cable modem interrupts.

Figure 7 Sample Debug Cable-Modem Interrupts Output

```
uBR904# debug cable-modem interrupts
*** bcm3220_rx_mac_msg_interrupt ***
*** bcm3220_rx_mac_msg_interrupt ***
### bcm3220_tx_interrupt ###
*** bcm3220_rx_mac_msg_interrupt ***
### bcm3220_tx_interrupt ###
*** bcm3220_rx_mac_msg_interrupt ***
### bcm3220_tx_interrupt ###
### bcm3220_tx_interrupt ###
### bcm3220_tx_interrupt ###
### bcm3220_tx_interrupt ###
```

Related Commands

debug cable-modem bpkm
debug cable-modem bridge
debug cable-modem error
debug cable-modem mac
debug cable-modem map

debug cable-modem mac

Use the **debug cable-modem mac** Privileged EXEC command to troubleshoot the cable modem MAC layer. The **no** form of this command turns debugging messages off.

[**no**] **debug cable-modem mac** {**log** [**verbose**] | **messages**}

Syntax Description

log	Realtime MAC log display.
verbose	(Optional) Displays periodic MAC layer events, such as ranging.
messages	MAC layer management messages.

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Of all the available debug cable modem commands, the most useful is **debug cable-modem mac log**.

Mac log messages are written to a circular log file even when debugging is not turned on. These messages include timestamps, events, and information pertinent to these events. Enter the **debug cable-modem mac log** command to view Mac log messages. If you want to view this information without entering debug mode, enter the **show controller cable-modem** *number* **mac log** command. The same information is displayed by both commands.

If the cable modem interface fails to come up or resets periodically, the Mac log will show what happened. For example, if an address is not obtained from the DHCP server, an error is logged, initialization starts over, and the cable modem scans for a downstream frequency. The **debug cable-modem mac log** command displays the log from oldest entry to newest entry.

After initial ranging is successful (dhcp_state has been reached), further `RNG-REQ/RNG-RSP` messages and watchdog timer entries are suppressed from output, unless the **verbose** keyword suffix is used. For example, the **debug cable-modem mac log verbose** command is entered. Note that `CMAC_LOG_WATCHDOG_TIMER` entries while in the maintenance_state are normal when using the **verbose** keyword.

Sample Displays

Figure 8 shows sample display output from the **debug cable-modem mac log** Privileged EXEC command. After the debug command is entered, the fields of the output are: the date, local time, seconds since bootup, the log message, and in some cases a parameter that gives more detail about the log entry.

The line "`0 events dropped due to lack of a chunk`" at the end of a display indicates that no log entries were discarded due to a temporary lack of memory. This means the log is accurate and reliable.

Figure 8 Sample Debug Cable-Modem Mac Log Output

```
uBR904# debug cable-modem mac log
*Mar  7 01:42:59: 528302.040 CMAC_LOG_LINK_DOWN
*Mar  7 01:42:59: 528302.042 CMAC_LOG_RESET_FROM_DRIVER
*Mar  7 01:42:59: 528302.044 CMAC_LOG_STATE_CHANGE                    wait_for_link_up_state
*Mar  7 01:42:59: 528302.046 CMAC_LOG_DRIVER_INIT_IDB_SHUTDOWN        0x08098D02
*Mar  7 01:42:59: 528302.048 CMAC_LOG_LINK_DOWN
*Mar  7 01:43:05: 528308.428 CMAC_LOG_DRIVER_INIT_IDB_RESET           0x08098E5E
*Mar  7 01:43:05: 528308.432 CMAC_LOG_LINK_DOWN
*Mar  7 01:43:05: 528308.434 CMAC_LOG_LINK_UP
*Mar  7 01:43:05: 528308.436 CMAC_LOG_STATE_CHANGE                    ds_channel_scanning_state
*Mar  7 01:43:05: 528308.440 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   88/453000000/855000000/6000000
*Mar  7 01:43:05: 528308.444 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   89/93000000/105000000/6000000
*Mar  7 01:43:05: 528308.448 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   90/111250000/117250000/6000000
*Mar  7 01:43:05: 528308.452 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   91/231012500/327012500/6000000
*Mar  7 01:43:05: 528308.456 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   92/333015000/333015000/6000000
*Mar  7 01:43:05: 528308.460 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   93/339012500/399012500/6000000
*Mar  7 01:43:05: 528308.462 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   94/405000000/447000000/6000000
*Mar  7 01:43:05: 528308.466 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   95/123015000/129015000/6000000
*Mar  7 01:43:05: 528308.470 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   96/135012500/135012500/6000000
*Mar  7 01:43:05: 528308.474 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   97/141000000/171000000/6000000
*Mar  7 01:43:05: 528308.478 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   98/219000000/225000000/6000000
*Mar  7 01:43:05: 528308.482 CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   99/177000000/213000000/6000000
*Mar  7 01:43:05: 528308.486 CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY  663000000
*Mar  7 01:43:05: 528308.488 CMAC_LOG_WILL_SEARCH_USER_DS_FREQUENCY   663000000
*Mar  7 01:43:07: 528310.292 CMAC_LOG_DS_64QAM_LOCK_ACQUIRED          663000000
.
.
.
528383.992 CMAC_LOG_STATE_CHANGE                      registration_state
528384.044 CMAC_LOG_REG_REQ_MSG_QUEUED
528384.050 CMAC_LOG_REG_REQ_TRANSMITTED
528384.052 CMAC_LOG_REG_RSP_MSG_RCVD
528384.078 CMAC_LOG_COS_ASSIGNED_SID                  1/4
528384.102 CMAC_LOG_RNG_REQ_QUEUED                    4
528384.102 CMAC_LOG_REGISTRATION_OK
528384.102 CMAC_LOG_STATE_CHANGE                      establish_privacy_state
528384.102 CMAC_LOG_STATE_CHANGE                      maintenance_state
528388.444 CMAC_LOG_RNG_REQ_TRANSMITTED
528388.444 CMAC_LOG_RNG_RSP_MSG_RCVD
528398.514 CMAC_LOG_RNG_REQ_TRANSMITTED
528398.516 CMAC_LOG_RNG_RSP_MSG_RCVD
528408.584 CMAC_LOG_RNG_REQ_TRANSMITTED
528408.586 CMAC_LOG_RNG_RSP_MSG_RCVD
528414.102 CMAC_LOG_WATCHDOG_TIMER
528418.654 CMAC_LOG_RNG_REQ_TRANSMITTED
528418.656 CMAC_LOG_RNG_RSP_MSG_RCVD
528428.726 CMAC_LOG_RNG_REQ_TRANSMITTED
528428.728 CMAC_LOG_RNG_RSP_MSG_RCVD
528438.796 CMAC_LOG_RNG_REQ_TRANSMITTED
528438.798 CMAC_LOG_RNG_RSP_MSG_RCVD
528444.102 CMAC_LOG_WATCHDOG_TIMER
528444.492 CMAC_LOG_LINK_DOWN
528444.494 CMAC_LOG_RESET_FROM_DRIVER
528444.494 CMAC_LOG_STATE_CHANGE                      wait_for_link_up_state
528444.494 CMAC_LOG_DRIVER_INIT_IDB_SHUTDOWN          0x08098D02
528444.494 CMAC_LOG_LINK_DOWN
528474.494 CMAC_LOG_WATCHDOG_TIMER
528504.494 CMAC_LOG_WATCHDOG_TIMER
528534.494 CMAC_LOG_WATCHDOG_TIMER

0 events dropped due to lack of a chunk
```

Figure 9 compares the output of the **debug cable-modem mac log** command with the **debug cable-modem mac log verbose** command. The keyword **verbose** displays periodic events such as ranging.

Figure 9 Sample Debug Cable-Modem Mac Log and Verbose Output

```
uBR904# debug cable mac log
Cable Modem mac log debugging is on
uBR904#
uBR904#
uBR904# debug cable mac log verbose
Cable Modem mac log debugging is on (verbose)
uBR904#
574623.810 CMAC_LOG_RNG_REQ_TRANSMITTED
574623.812 CMAC_LOG_RNG_RSP_MSG_RCVD
574627.942 CMAC_LOG_WATCHDOG_TIMER
574633.880 CMAC_LOG_RNG_REQ_TRANSMITTED
574633.884 CMAC_LOG_RNG_RSP_MSG_RCVD
574643.950 CMAC_LOG_RNG_REQ_TRANSMITTED
574643.954 CMAC_LOG_RNG_RSP_MSG_RCVD
574654.022 CMAC_LOG_RNG_REQ_TRANSMITTED
574654.024 CMAC_LOG_RNG_RSP_MSG_RCVD
574657.978 CMAC_LOG_WATCHDOG_TIMER
574664.094 CMAC_LOG_RNG_REQ_TRANSMITTED
574664.096 CMAC_LOG_RNG_RSP_MSG_RCVD
574674.164 CMAC_LOG_RNG_REQ_TRANSMITTED
574674.166 CMAC_LOG_RNG_RSP_MSG_RCVD
uBR904# no debug cable mac log verbose
Cable Modem mac log debugging is off
uBR904#
574684.234 CMAC_LOG_RNG_REQ_TRANSMITTED
574684.238 CMAC_LOG_RNG_RSP_MSG_RCVD
```

Figure 10 shows display output for the **debug cable mac messages** Privileged EXEC command.

This command causes received cable MAC management messages to be displayed in a verbose format. The messages that are displayed are UCD, MAP, RNG-RSP, REG-RSP and UCC. In addition, transmitted REG-REQs are displayed in hex dump format. The output from this command is very verbose and is usually not needed for normal interface debugging. The command is most useful when attempting to attach a cable modem to an uncertified CMTS. For a description of the displayed fields of each message, refer to the MCNS DOCSIS RFI spec, v1.0.

Figure 10 Sample Debug Cable-Modem Mac Messages Output

```
uBR904# debug cable mac messages
*Mar  7 01:44:06:
*Mar  7 01:44:06: UCD MESSAGE
*Mar  7 01:44:06: -----------
*Mar  7 01:44:06:   FRAME HEADER
*Mar  7 01:44:06:     FC                          - 0xC2 == MAC Management
*Mar  7 01:44:06:     MAC_PARM                    - 0x00
*Mar  7 01:44:06:     LEN                         - 0xD3
*Mar  7 01:44:06:   MAC MANAGEMENT MESSAGE HEADER
*Mar  7 01:44:06:     DA                          - 01E0.2F00.0001
*Mar  7 01:44:06:     SA                          - 00E0.1EA5.BB60
*Mar  7 01:44:06:     msg LEN                     - C1
*Mar  7 01:44:06:     DSAP                        - 0
*Mar  7 01:44:06:     SSAP                        - 0
```

```
*Mar  7 01:44:06:     control                   - 03
*Mar  7 01:44:06:     version                   - 01
*Mar  7 01:44:06:     type                      - 02 == UCD
*Mar  7 01:44:06:     RSVD                      - 0
*Mar  7 01:44:06:   US Channel ID               - 1
*Mar  7 01:44:06:   Configuration Change Count  - 4
*Mar  7 01:44:06:   Mini-Slot Size              - 8
*Mar  7 01:44:06:   DS Channel ID               - 1
*Mar  7 01:44:06:   Symbol Rate                 - 8
*Mar  7 01:44:06:   Frequency                   - 20000000
*Mar  7 01:44:06:   Preamble Pattern           - CC CC CC CC CC CC CC CC CC CC CC CC CC
CC 0D 0D
*Mar  7 01:44:06:   Burst Descriptor 0
*Mar  7 01:44:06:     Interval Usage Code       - 1
*Mar  7 01:44:06:     Modulation Type           - 1 == QPSK
*Mar  7 01:44:06:     Differential Encoding     - 2 == OFF
*Mar  7 01:44:06:     Preamble Length           - 64
*Mar  7 01:44:06:     Preamble Value Offset     - 56
*Mar  7 01:44:06:     FEC Error Correction      - 0
*Mar  7 01:44:06:     FEC Codeword Info Bytes   - 16
*Mar  7 01:44:06:     Scrambler Seed            - 0x0152
*Mar  7 01:44:06:     Maximum Burst Size        - 1
*Mar  7 01:44:06:     Guard Time Size           - 8
*Mar  7 01:44:06:     Last Codeword Length      - 1 == FIXED
*Mar  7 01:44:06:     Scrambler on/off          - 1 == ON
*Mar  7 01:44:06:   Burst Descriptor 1
*Mar  7 01:44:06:     Interval Usage Code       - 3
*Mar  7 01:44:06:     Modulation Type           - 1 == QPSK
*Mar  7 01:44:06:     Differential Encoding     - 2 == OFF
*Mar  7 01:44:06:     Preamble Length           - 128
*Mar  7 01:44:06:     Preamble Value Offset     - 0
*Mar  7 01:44:06:     FEC Error Correction      - 5
*Mar  7 01:44:06:     FEC Codeword Info Bytes   - 34
*Mar  7 01:44:06:     Scrambler Seed            - 0x0152
*Mar  7 01:44:06:     Maximum Burst Size        - 0
*Mar  7 01:44:06:     Guard Time Size           - 48
*Mar  7 01:44:06:     Last Codeword Length      - 1 == FIXED
*Mar  7 01:44:06:     Scrambler on/off          - 1 == ON
*Mar  7 01:44:06:   Burst Descriptor 2
*Mar  7 01:44:06:     Interval Usage Code       - 4
*Mar  7 01:44:06:     Modulation Type           - 1 == QPSK
*Mar  7 01:44:06:     Differential Encoding     - 2 == OFF
*Mar  7 01:44:06:     Preamble Length           - 128
*Mar  7 01:44:06:     Preamble Value Offset     - 0
*Mar  7 01:44:06:     FEC Error Correction      - 5
*Mar  7 01:44:06:     FEC Codeword Info Bytes   - 34
*Mar  7 01:44:06:     Scrambler Seed            - 0x0152
*Mar  7 01:44:06:     Maximum Burst Size        - 0
*Mar  7 01:44:06:     Guard Time Size           - 48
*Mar  7 01:44:06:     Last Codeword Length      - 1 == FIXED
*Mar  7 01:44:06:     Scrambler on/off          - 1 == ON
*Mar  7 01:44:06:   Burst Descriptor 3
*Mar  7 01:44:06:     Interval Usage Code       - 5
*Mar  7 01:44:06:     Modulation Type           - 1 == QPSK
*Mar  7 01:44:06:     Differential Encoding     - 2 == OFF
*Mar  7 01:44:06:     Preamble Length           - 72
*Mar  7 01:44:06:     Preamble Value Offset     - 48
*Mar  7 01:44:06:     FEC Error Correction      - 5
*Mar  7 01:44:06:     FEC Codeword Info Bytes   - 75
*Mar  7 01:44:06:     Scrambler Seed            - 0x0152
*Mar  7 01:44:06:     Maximum Burst Size        - 0
*Mar  7 01:44:06:     Guard Time Size           - 8
*Mar  7 01:44:06:     Last Codeword Length      - 1 == FIXED
*Mar  7 01:44:06:     Scrambler on/off          - 1 == ON
*Mar  7 01:44:06:
```

```
*Mar  7 01:44:06:
*Mar  7 01:44:06: MAP MESSAGE
*Mar  7 01:44:06: -----------
*Mar  7 01:44:06:   FRAME HEADER
*Mar  7 01:44:06:    FC                     - 0xC3 == MAC Management with Extended Header
*Mar  7 01:44:06:     MAC_PARM                  - 0x02
*Mar  7 01:44:06:     LEN                       - 0x42
*Mar  7 01:44:06:     EHDR                      - 0x00 0x00
*Mar  7 01:44:06:   MAC MANAGEMENT MESSAGE HEADER
*Mar  7 01:44:06:     DA                        - 01E0.2F00.0001
.
.
.
*Mar  7 01:44:17: RNG-RSP MESSAGE
*Mar  7 01:44:17: ---------------
*Mar  7 01:44:17:   FRAME HEADER
*Mar  7 01:44:17:     FC                        - 0xC2 == MAC Management
*Mar  7 01:44:17:     MAC_PARM                  - 0x00
*Mar  7 01:44:17:     LEN                       - 0x2B
*Mar  7 01:44:17:   MAC MANAGEMENT MESSAGE HEADER
*Mar  7 01:44:17:     DA                        - 00F0.1EB2.BB61
.
.
.
*Mar  7 01:44:20: REG-REQ MESSAGE
*Mar  7 01:44:20: ---------------
*Mar  7 01:44:20: C20000A5 000000E0  1EA5BB60 00F01EB2
*Mar  7 01:44:20: BB610093 00000301  06000004 03010104
*Mar  7 01:44:20: 1F010101 0204003D  09000304 001E8480
*Mar  7 01:44:20: 04010705 04000186  A0060200 0C070101
*Mar  7 01:44:20: 080300F0 1E112A01  04000000 0A020400
*Mar  7 01:44:20: 00000A03 04000002  58040400 00000105
*Mar  7 01:44:20: 04000000 01060400  00025807 04000000
*Mar  7 01:44:20: 3C2B0563 6973636F  06105E4F C908C655
*Mar  7 01:44:20: 61086FD5 5C9D756F  7B730710 434D5453
*Mar  7 01:44:20: 204D4943 202D2D2D  2D2D2D2D 0C040000
*Mar  7 01:44:20: 00000503 010100
*Mar  7 01:44:20:
*Mar  7 01:44:20:
*Mar  7 01:44:20: REG-RSP MESSAGE
*Mar  7 01:44:20: ---------------
*Mar  7 01:44:20:   FRAME HEADER
*Mar  7 01:44:20:     FC                        - 0xC2 == MAC Management
*Mar  7 01:44:20:     MAC_PARM                  - 0x00
*Mar  7 01:44:20:     LEN                       - 0x29
*Mar  7 01:44:20:   MAC MANAGEMENT MESSAGE HEADER
*Mar  7 01:44:20:     DA                        - 00F0.1EB2.BB61
.
.
.
```

Related Commands

debug cable-modem bpkm
debug cable-modem bridge
debug cable-modem error
debug cable-modem interrupts
debug cable-modem map

debug cable-modem map

Use the **debug cable-modem map** Privileged EXEC command to display the timing from MAP messages to sync messages and the timing between MAP messages. The **no** form of this command disables debugging output.

[**no**] **debug cable-modem map**

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3 NA.

Sample Display

Figure 11 shows display output for the **debug cable map** Privileged EXEC command.

Figure 11 Sample Debug Cable-Modem Map Output

```
uBR904# debug cable-modem map
Cable Modem MAP debugging is on
uBR904#
*Mar  7 20:12:08: 595322.942: Min MAP to sync=72
*Mar  7 20:12:08: 595322.944: Max map to map time is 40
*Mar  7 20:12:08: 595322.982: Min MAP to sync=63
*Mar  7 20:12:08: 595323.110: Max map to map time is 41
*Mar  7 20:12:08: 595323.262: Min MAP to sync=59
*Mar  7 20:12:08: 595323.440: Max map to map time is 46
*Mar  7 20:12:09: 595323.872: Min MAP to sync=58
```

Related Commands

debug cable-modem bpkm
debug cable-modem bridge
debug cable-modem error
debug cable-modem interrupts
debug cable-modem mac

What to do Next

For more troubleshooting tips, see the chapter "Troubleshooting the Installation" in the *Cisco Cable Modem Installation and Configuration Guide*.