DOCSIS Configuration

This chapter identifies those functions you must check or configure to bring a BACC DOCSIS deployment into operation. It also provides information required prior to configuration, and describes the available tools.

- DOCSIS Workflow, page 4-2
- Using MIBs with Dynamic DOCSIS Templates, page 4-4
- BACC Features for DOCSIS Configurations, page 4-5
- Troubleshooting DOCSIS Networks, page 4-6
- Network Layer Configuration and Above, page 4-7
- Troubleshooting Cable Modem States, page 4-10

Note See the "DOCSIS Option Support" section on page 12-12 for information concerning the DOCSIS options supported by this BACC release.

DOCSIS Workflow

The provisioning (workflow) steps contained in the DOCSIS Provisioning Specification are shown in Figure 4-1.

Figure 4-1 DOCSIS Provisioning Flow

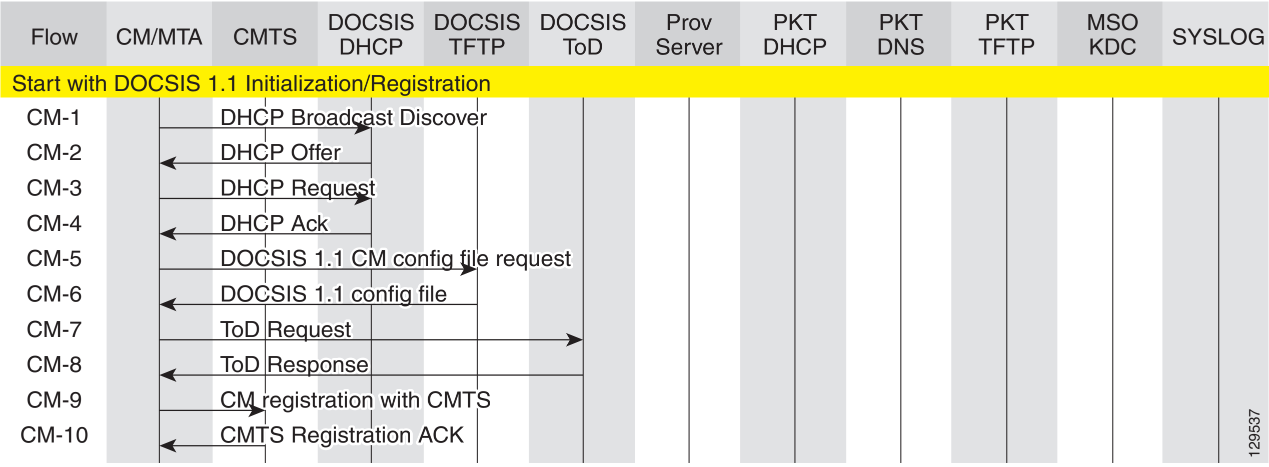

Table 4-1 describes the potential problems that can exist at various DOCSIS provisioning steps illustrated in Figure 4-1.

Table 4-1 DOCSIS Workflow Description

Step	DOCSIS Workflow step	Potential Problems
CM-1	DHCP Discover	• The init(d) state • No addresses available • Verify BACC shared-secret is correct • Incorrectly configured Class of Service • DOCSIS template parsing errors (invalid option, include file - not found, and so on) **CNR DHCP** • Incorrect CNR DHCP configuration • Verify DHCP server is in the provisioning group **BACC CNR Extension** • BACC CNR extension cannot contact DPEs • BACC CNR extension fails to find any DPEs in provisioning group • Verify extensions are connected to the RDU • BACC CNR extension gets DPE cache miss, sends request to RDU **RDU** • No appropriate scopes defined (or not match BACC RDU configuration) • Verify RDU IP address is correct • Verify RDU port is correct (default 49187) • Verify the RDU can be pinged from the DPE • Configuration generation is failing at the RDU • RDU licenses exceeded, not configured • Device detection is failing at the RDU **DPE** • Verify DPEs assigned to the provisioning group • Verify DPEs can be pinged from the DHCP server • Verify DPE interfaces enabled for provisioning
CM-2	DHCP Offer	Routing issues between DHCP and CMTS
CM-3	DHCP Request	init(i) state DHCP server did not provide all the parameters required
CM-4	DHCP Ack	

Table 4-1 DOCSIS Workflow Description (continued)

Step	DOCSIS Workflow step	Potential Problems
CM-5	TFTP Request	• init(o) state • routing issues between CMTS and DPE • No route from TFTP server (DPE) to modem • DPE cache miss (static file, and RDU down or doesn't have file) • File not found at TFTP server (DPE) • DPE cache miss (dynamic file) • DPE IP validation failure (for example, the IP address of the device is not what was expected, the Dynamic Shared Secret is enabled on CMTS, or a hacker is spoofing as a DOCSIS modem.)
CM-6	TFTP Response	Routing issues between DPE and CMTS
CM-7	ToD Request	init(t) state - No route from time server (DPE) to Modem
CM-8	ToD Response	
CM-9	CM register with CMTS	• reject(m) - * CMTS shared secret mismatch with BACC or DPE DSS shared secret • reject(c) - * delivered incorrect DOCSIS configuration file (1.1 file to 1.0 CM)
CM-10	CMTS registration Ack	Acceptable states are: • online • online(d) • online(pk) • online(pt)

Using MIBs with Dynamic DOCSIS Templates

For a full list of MIBs shipped with BACC, see SNMP Varbind, page 12-5.

Two versions of the DOCSIS MIB are loaded into the RDU:

- DOCS-CABLE-DEVICE-MIB-OBSOLETE (experimental branch)
- DOCS-CABLE-DEVICE-MIB (mib2 branch)

For information on how to use them, see DOCSIS MIBs, page 12-5.

You can add MIBs using an API call or by modifying rdu.properties. For more details, see Configuring Euro-PacketCable MIBs, page 5-6.

You can add a SNMP TLV to a template when no MIB is available. See Adding SNMP TLVs Without a MIB, page 12-10 for more information.

BACC Features for DOCSIS Configurations

This section describes BACC value-added features as they relate to the DOCSIS technology.

Dynamic Configuration TLVs

These TLVs are added by the DPE whenever a TFTP request for dynamic DOCSIS configuration is received:

- TLV 19: TFTP Server Timestamp (optional)—Displayed in the Configure DOCSIS Defaults page as the TFTP Time Stamp Option. See Table 10-5 on page 10-14 for more information. This TLV requires NTP synchronization on CMTS and DPE.
- TLV 20: TFTP Server Provisioned Modem Address (optional)—Displayed in the Configure DOCSIS Defaults page as the TFTP Modem Address Option. See Table 10-5 on page 10-14 for more information.
- TLV 6: CM MIC Configuration Setting (required)
- TLV 7: CMTS MIC Configuration Setting (required)—Displayed in the Configure DOCSIS Defaults page as the CMTS Shared Secret. See Table 10-5 on page 10-14 for more information.

Note When configuring CMTS MIC, note the following CMTS IOS release dependencies:

- DOCSIS 2.0 CMTS MIC requires CMTS IOS 12.3BC when including TLV 39 or TLV 40.
- Certain CMTS IOS commands are assumed to be configured by BACC:
  - ip dhcp relay information option
  - no ip dhcp relay information check
  - cable helper-address <x.x.x.x>
  - cable dhcp-giaddr primary

DPE TFTP IP Validation

For dynamic configuration files, the DPE TFTP server verifies that the IP address of the TFTP client matches the expected DOCSIS cable modem IP address. If it does not match, the request is dropped. This feature is incompatible with the Cisco CMTS DMIC feature.

Use the no tftp verify-ip command to disable the verification of requestor IP addresses on dynamic configuration TFTP requests:

```
dpe# no tftp verify-ip
%OK
```

.DOCSIS 1.0, 1.1, 2.0 Support

BACC 2.7 supports DOCSIS 1.0, 1.1, and 2.0. See Chapter 12, "Broadband Access Center for Cable Support Tools and Advanced Concepts" for additional information describing the TLVs and options supported in each DOCSIS version by BACC.

Dynamic DOCSIS Version Selection

BACC can detect a cable modem's DOCSIS version from an incoming DHCP request. It can also detect the CMTS DOCSIS version from any customer-supplied source that provides a mapping of GIADDR to DOCSIS version numbers.

You can specify the DOCSIS version in the Configuration File utility. See Using the Configuration File Utility, page 12-25 for more information.

DOCSIS Configuration File Based on DOCSIS Version

The default DOCSIS code generation extension uses the version of DOCSIS for provisioning the modem, to determine the filename of the DOCSIS configuration file to send to the modem.

DOCSIS Version Dependent Class of Service Properties

The following new class of service properties are supported by the Administrator's user interface and by the BACC API:

```
/cos/docsis/file/1.0
/cos/docsis/file/1.1
/cos/docsis/file/2.0
```

These properties can optionally be added to a DOCSIS class of service to associate a DOCSIS configuration filename with a particular DOCSIS version. Each of these properties (if set) causes the RDU to establish a database relationship between the class of service and the file named by the property value, as is done for the existing DOCSIS configuration filename property.

Troubleshooting DOCSIS Networks

DOCSIS provides the bandwidth and latency guarantees needed to provide toll-quality voice, data services, and multimedia applications across a shared HFC network. It is designed to be backward compatible, enabling DOCSIS 1.0, 1.1, and 2.0 modems to operate in the same spectrum on the same network.

Note The bulk of this section discusses the DOCSIS technology with respect to BACC and the Cisco uBR7246 CMTS. The examples are for the Cisco uBR7246 CMTS; however the content is relative to the DOCSIS standard, which other CMTS vendors must support.

Network Layer Configuration and Above

After ranging is successful, the cable modem needs additional configuration from the operational support servers. The collection of services they provide and configure are called operational support services (OSS).

Note At this point, the modem is functional; it could act like a bridge and be a fairly quick transmission device, but there would be no administrator configuration, diagnosis, class of services, security, privacy, or remote software upgrade capability, *and therefore* it would not be DOCSIS compliant.

Establish DHCP State

```
MAC State --->>> 'establish_dhcp_state'
cmWriteFlashFile("CM_MACCONFIG", 0x8313db6, 0x12e) by TID 0x82eb0f8 (tConfigNV)
usrEraseSysFlash(1, 0x1e0400, 0x12e) by TID 0x82eb0f8 (tConfigNV)
.........cmWriteFlashFile("CM_DHCPLEASE", 0x810ebe4, 0x24) by TID 0x82d825c (tLease-0)
usrEraseSysFlash(1, 0x1e0200, 0x24) by TID 0x82d825c (tLease-0)
```

The first task in configuring the OSS is acquiring network configuration via DHCP. A broadcast DHCP DISCOVER message is issued by the modem. Normally, routers will not forward broadcast messages, but they can be configured to do so for a collection of UDP protocols, one of which is DHCP. If a DHCP server responds to the DISCOVER with an OFFER, the modem may choose to send a REQUEST for the offered configuration.

The DHCP server can respond with an ACK (acknowledged) or NAK (not acknowledged). A NAK may be the result of an incompatible IP address and gateway address, which might occur if a modem hopped from one downstream channel to another that resides on a different subnet. When the modem seeks renewal of the lease, the IP address and the gateway address of the DHCP REQUEST message will be different network numbers and the DHCP server will refuse the REQUEST with a NAK. These situations are rare and the modem will simply release the lease and start over with a DHCP DISCOVER message.

Frequently, errors at establish_dhcp_state manifest as timeouts rather than NAKs. To troubleshoot effectively:

- Start at the lower layers and work up.
- Start locally and work towards remote possibilities.
- Look at the modem, before diagnosing the DHCP server via DHCP logs.

The log mask 0xffff00bf used above in ranging can also help you to troubleshoot almost all the problems a modem might encounter in bring-up stage and is a good place to start when troubleshooting a bring-up error. The order of DHCP messages should be DISCOVER, OFFER, REQUEST, ACK. If the modem is transmitting a DISCOVER with no OFFER response from the DHCP server, turn on UDP debugging on the uBR, using the following command:

uBR# **debug ip udp**

Note If connected to the router via a telnet session or any session other than a console session, the **terminal monitor** command (abbreviated **term mon**) is necessary to view debug messages. To turn off **term mon**, use the **term no mon** command or simply logout.

Caution Running debug commands on a uBR7246 with more than a few modems may cause the uBR7246 to halt the system in order to keep up with the debugging. In this case, all the modems may lose sync and debugging will be useless.

If there are no packets in the debug messages, check the configuration of the "ip helper-address" statement on the cable interface to which this modem is attached. If this is configured correctly and a packet trace of the DHCP server subnet also reveals no DHCP packets from the modem, then look at the output errors of the modem's cable interface, or the input errors of the cable interface of the uBR. It might be a good idea to boost the transmitter power a bit more with more attenuation.

If packets are being transmitted onto the DHCP server subnet, double-check the modem debug messages to see if there are parameter request or assignment errors. At this stage of troubleshooting you should investigate the routing between the modem and the DHCP server. It is also advisable to double-check the DHCP server configuration.

Establishing the ToD State

```
MAC State --->>> 'establish_tod_state'
```

After a modem has acquired its network parameters, it must request the time of day from a Time Of Day (TOD) server. TOD uses a UTC timestamp (seconds from January 1, 1970) combined with the time offset option value from DHCP to calculate the current time. The time is used for syslog and event log timestamps.

Time of day errors almost always point to a DHCP misconfiguration. Possible misconfigurations that can result in TOD errors are IP address, gateway address misconfigurations, or the wrong TOD server address. Again, start troubleshooting locally by examining the DHCP parameters the modem has stored, using the following command:

```
-> cmShowDhcpParameters
```

or:

```
-> cmAddLogValues ("SEV_ALL FAC_DHCP")
```

The latter will give debug output of DHCP actions and tasks.

Also check the modem routing table with the vxWorks **routeShow** command.

Security Association State

```
MAC State --->>> 'security_association_state'
```

This is a placeholder for a state yet to be defined. It is envisioned that a security association with a security server will provide IPsec-like security for the modems.Its design and implementation are still being discussed. In the meantime, DOCSIS 1.0 requires modems to support a future definition of this state including the DHCP option for a security server. It is unlikely to find a modem having a problem with this state until it is defined in DOCSIS and implemented on the modem.

Configuration File State

The main configuration and administration interface to the CM is the configuration file downloaded from the provisioning server. This configuration file contains downstream channel and upstream channel identification and characteristics, as well as class of service settings, baseline privacy settings, general operational settings, network management information, software upgrade fields, filters, and vendor specific settings.

Common reasons for failure at this state are missing file, wrong file permissions, TFTP server is unreachable, file is wrong format, file has missing required options, misconfigured required options, or incorrect options, for example, unknown or invalid TLVs.

A debug command for configuration file transactions and parsing is:

```
-> cmAddLogValues ("SEV_ALL FAC_CONFIGFILE")
```

Registration State

```
MAC State --->>> 'registration_state'
```

After configuration, the modem sends a registration request (REG-REQ) with a required subset of the configuration settings, as well as the CM and CMTS message integrity checks (MIC). The CM MIC is a hashed calculation over the configuration file settings which provides a method for the modem to be sure the configuration file was not tampered with in transit. The CMTS MIC is similar but also includes a setting for a shared-secret authentication string. This shared secret is known by the CMTS and the provisioning server, and ensures that only modems configured by authenticated provisioning servers will be allowed to register with the CMTS.

Problems with registration state almost always point to a configuration file error. Make sure the modem and the CMTS both support the settings in the configuration file. Make sure the CMTS allows the creation of class of service profiles or use a profile created by the CMTS. Check the authentication strings in the CMTS cable interface configuration and the configuration file.

Establish Privacy State

```
MAC State --->>> 'establish_privacy_state'
```

The modem must negotiate baseline privacy with the CMTS through the Baseline Privacy Key Management protocol, if all of the following are true:

- The modem software supports baseline privacy.
- The class of service is a privacy enabled profile.
- Baseline privacy settings are present in the configuration file.

If errors occur in this state, the likely causes are configuration file misconfigurations. Be certain that both the CM and the CMTS support baseline privacy and are enabled in the interface configuration for the CMTS and the configuration file for the CM.

You might also encounter errors due to encryption export restrictions, some vendor modems may require the following command on the uBR7246 in the interface configuration:

```
uBR(config-if)# cable privacy 40-bit-des
```

Operational State

```
MAC State --->>> 'operational_state' cmWriteFlashFile("CM_BOOT", 0x8109180, 0x13e) by TID
0x83049b8 (tMACCtrl)
usrEraseSysFlash(1, 0x1e0900, 0x13e) by TID 0x83049b8 (tMACCtrl)
```

If registration and baseline privacy negotiation (if required) succeed, the modem is operational and is ready to pass traffic.

A modem may become operational but not remain in operational state. This could be caused by sync loss or DHCP lease renewal failure, for example.

Troubleshooting Cable Modem States

This section discusses these cable modem states:

- Online State, page 4-10
- Offline State, page 4-11
- Ranging Process - init(r1),init(r2), and init(rc) state, page 4-16
- DHCP - init(d) state, page 4-18
- DHCP - init(i) state, page 4-20
- TOD exchange- init(t) state, page 4-23
- Option File Transfer Started-init(o) State, page 4-25
- Online, Online(d), Online(pk), Online(pt) state, page 4-27
- Reject(pk) and Reject(pt) state, page 4-30
- Registration - reject (m) state, page 4-31
- Registration - reject (c) state, page 4-32

Online State

This section provides information on understanding the *online state* of the cable modem, and troubleshooting problems indicated by this state, including RF problems.

Note Many but not all of the problems addressed in this section are RF-related. You should review this section to generally understand troubleshooting the problems between the Cisco uBR7246VXR and the CMs in its domain.

The first and most useful command to use at the Cisco uBR7246VXR is **show cable modem**:

#**show cable modem**

```
Interface Prim      Online        Timing    Rec     QoS    CPE    IP address    MAC address
Sid                 State         Offset    Power
Cable2/0/U0  5      online(pt)    2290       -0.25  5      0      10.1.1.25     0050.7366.2223
Cable2/0/U0  6      offline       2287      -0.25   2      0      10.1.1.26     0050.7366.2221
```

The Online State field identifies the status. Table 4-2 displays the possible values for the state.

Table 4-2 State Values

State Value	Description
offline	Cable modem considered offline
init (r1)	Cable modem sent initial ranging
init (r2)	Cable modem is ranging
init (rc)	Cable modem ranging complete
init (d)	DHCP request received
init (i)	DHCP reply received; IP address assigned
init (t)	TOD exchange started
init (o)	Option file transfer started
online	Cable modem registered, enabled for data
online(d)	Cable modem registered, but network access for the cable modem is disabled
online(pk)	Cable modem registered, BPI enabled and KEK assigned
online(pt)	Cable modem registered, BPI enabled and TEK assigned
reject (pk)	KEK modem key assignment rejected
reject (pt)	TEK modem key assignment rejected
reject (m)	Cable modem did attempt to register; registration was refused due to bad MIC (Message Integrity Check)
reject (c)	Cable modem did attempt to register; registration was refused due to bad COS (Class of Service)

The following sections discuss each *State* value, the possible causes, and the steps that can be taken to arrive at the correct state (online).

Offline State

In some cases the cable modem may cycle through other states then back to offline. The following list gives the most common reasons for a cable modem not to achieve QAM lock:

- Weak carrier signal (too much noise).
- Incorrect downstream center frequency.
- Incorrect frequency specified in the DOCSIS file.
- Absence of downstream digital QAM modulated signal.
- Incorrect frequency specified in **cable modem change-frequency** on the Cisco uBR7246VXR.

The following is a portion of the output from **show controllers cable-modem 0** entered at the modem:

```
#sh cont c 0

BCM Cable interface 0:
CM unit 0, idb 0x8086C88C, ds 0x8086E460, regaddr = 0x2700000, reset_mask 0x80
station address 0030.96f9.65d9  default station address 0030.96f9.65d9
PLD VERSION: 1
```

```
Concatenation: ON Max bytes Q0: 2000 Q1: 2000 Q2: 2000 Q3: 2000

MAC State is ds_channel_scanning_state, Prev States = 3
MAC mcfilter 01E02F00  data mcfilter 00000000

MAC extended header ON
DS: BCM 3300 Receiver: Chip id = BCM3300
US: BCM 3300 Transmitter: Chip id = 3300

Tuner: status=0x00
Rx: tuner_freq 529776400, symbol_rate 5361000, local_freq 11520000
    snr_estimate 166(TenthdB), ber_estimate 0, lock_threshold 26000
    QAM not in lock, FEC not in lock, qam_mode QAM_64    (Annex B)
Tx: tx_freq 27984000, symbol rate 8 (1280000 sym/sec)
    power_level: 6.0 dBmV (commanded)
        7  (gain in US AMP units)
        63  (BCM3300 attenuation in .4 dB units)
...[Rest of the displayed output is omitted]
```

The Signal to Noise ratio (SNR) estimate is 16.6 dB in the previous output example. Ideally this should be at least 30dB for the CM to operate properly for 64 QAM. See the RF specifications for DOCSIS Downstream and Upstream signals. In some cases you may have a good SNR (for example, 34dB) but still have noise present, such as impulse noise. This can be detected only by a spectrum analyzer operating in the zero span mode. One indication of impulse noise is errors that cannot be corrected. These are seen in the output of **show interfaces cable 2/0 upstream 0** as shown in the following output:

```
#show interfaces cable 2/0 upstream 0

    Cable2/0: Upstream 0 is up
    Received 46942 broadcasts, 0 multicasts, 205903 unicasts
    0 discards, 12874 errors, 0 unknown protocol
```

Note If a value is greater than 1 in 10,000, most likely impulse noise is present.

```
    252845 packets input, 1 uncorrectable
    12871 noise, 0 microreflections Total Modems On This Upstream Channel : 3 (3 active)
    Default MAC scheduler
    Queue[Rng Polls]  0/64, fifo queueing, 0 drops
    Queue[Cont Mslots]  0/104, fifo queueing, 0 drops
    Queue[CIR Grants]  0/64, fair queueing, 0 drops
    Queue[BE Grants]  0/64, fair queueing, 0 drops
    Queue[Grant Shpr]   0/64, calendar queueing, 0 drops
    Reserved slot table currently has 0 CBR entries
    Req IEs 77057520, Req/Data IEs 0
    Init Mtn IEs 1194343, Stn Mtn IEs 117174
    Long Grant IEs 46953, Short Grant IEs 70448
    Avg upstream channel utilization : 1%
    Avg percent contention slots : 96%
    Avg percent initial ranging slots : 4%
    Avg percent minislots lost on late MAPs : 0%
    Total channel bw reserved 0 bps
    CIR admission control not enforced
    Current minislot count   : 7192093    Flag: 0
    Scheduled minislot count : 7192182    Flag: 0
```

The optimal input power level at the CM is 0dBmV; the receiver has a range of -15dBmv to +15dBmV. This can be measured by the spectrum analyzer. If the power is too low you may need to configure the upconverter as described in the *Cisco uBR 7246 Hardware Installation Guide*. If the signal is too strong, you may need to add more attenuation at the high frequency port connection. If a particular frequency has too much noise present, you may need to select another frequency in the spectrum.

To confirm that the CM has not been able to achieve QAM lock, turn on **debug cable-modem mac log verbose**. You should see output similar to the following:

```
2d18h:   239198.516       CMAC_LOG_LINK_DOWN
2d18h:   239198.516   CMAC_LOG_LINK_UP
2d18h:   239198.516   CMAC_LOG_STATE_CHANGE                      ds_channel_scanning_state
2d18h:   239198.520   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   99/805790200/99770
2d18h:   239198.520   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   98/601780000/79970
2d18h:   239198.520   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   97/403770100/59570
2d18h:   239198.524   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   96/73753600/115750
2d18h:   239198.524   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   95/217760800/39770
2d18h:   239198.528   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   94/121756000/16970
2d18h:   239198.528   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   93/175758700/21170
2d18h:   239198.528   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   92/79753900/857540
2d18h:   239198.532   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   91/55752700/677530
2d18h:   239198.532   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   90/177000000/21300
2d18h:   239198.532   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   89/219000000/22500
2d18h:   239198.536   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   88/141000000/17100
2d18h:   239198.536   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   87/135012500/13500
2d18h:   239198.540   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   86/123012500/12900
2d18h:   239198.540   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   85/405000000/44700
2d18h:   239198.540   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   84/339012500/39900
2d18h:   239198.544   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   83/333025000/33300
2d18h:   239198.544   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   82/231012500/32700
2d18h:   239198.544   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   81/111025000/11700
2d18h:   239198.548   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   80/93000000/105000
2d18h:   239198.548   CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND   79/453000000/85500
NOTE—unable to lock on:
2d18h:   239198.552   CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY  453000000
2d18h:   239199.672   CMAC_LOG_DS_NO_QAM_FEC_LOCK              453000000
2d18h:   239200.788   CMAC_LOG_DS_NO_QAM_FEC_LOCK              453000000
2d18h:   239201.904   CMAC_LOG_DS_NO_QAM_FEC_LOCK              453000000
2d18h:   239203.020   CMAC_LOG_DS_NO_QAM_FEC_LOCK              459000000
```

Another reason the CM might not achieve QAM lock is if the incorrect downstream center frequency was configured on the upconverter. For example, on the NTSC frequency map for standard 6-MHz channel bands in North America, channel 100-100 uses 648.0-654.0 with a center frequency of 651 MHz. However, if you are using an upconverter that is designed to work with the video carrier frequency (for example, GI C6U which is 1.75MHz below the center frequency), you must set a frequency of 649.25 MHz for Channel 100-100.

Another common mistake is to specify an incorrect frequency value in the *Downstream Frequency* field under the Radio Frequency Info in the Cisco Broadband Configurator (available for download on cisco.com). Usually there is no need to specify a frequency value under this option. However, if there is a need (for example, when certain modems need to lock on a different frequency), you must select the proper frequency values, as explained previously. The following debug output illustrates this: a CM locks on initially at 453MHz and then at 535.25MHz (which was specified in the DOCSIS configuration file), thus causing the cable modem to reset and cycle through this process indefinitely:

```
4d00h:  345773.916  CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY     453000000
4d00h:  345774.956  CMAC_LOG_UCD_MSG_RCVD
4d00h:  345775.788  CMAC_LOG_DS_64QAM_LOCK_ACQUIRED             453000000
4d00h:  345775.792  CMAC_LOG_DS_CHANNEL_SCAN_COMPLETED
4d00h:  345775.794  CMAC_LOG_STATE_CHANGE                       wait_ucd_state
4d00h:  345776.946  CMAC_LOG_UCD_MSG_RCVD                       1
4d00h:  345778.960  CMAC_LOG_UCD_MSG_RCVD                       1
4d00h:  345778.962  CMAC_LOG_ALL_UCDS_FOUND
4d00h:  345778.966  CMAC_LOG_STATE_CHANGE                       wait_map_state
4d00h:  345778.968  CMAC_LOG_FOUND_US_CHANNEL                   1
4d00h:  345780.996  CMAC_LOG_UCD_MSG_RCVD                       1
4d00h:  345781.000  CMAC_LOG_UCD_NEW_US_FREQUENCY               27984000
4d00h:  345781.004  CMAC_LOG_SLOT_SIZE_CHANGED                  8
4d00h:  345781.084  CMAC_LOG_UCD_UPDATED
4d00h:  345781.210  CMAC_LOG_MAP_MSG_RCVD
4d00h:  345781.212  CMAC_LOG_INITIAL_RANGING_MINISLOTS          40
4d00h:  345781.216  CMAC_LOG_STATE_CHANGE                       ranging_1_state
4d00h:  345781.220  CMAC_LOG_RANGING_OFFSET_SET_TO              9610
4d00h:  345781.222  CMAC_LOG_POWER_LEVEL_IS                     22.0 dBmV (comma)
4d00h:  345781.226  CMAC_LOG_STARTING_RANGING
4d00h:  345781.228  CMAC_LOG_RANGING_BACKOFF_SET                0
4d00h:  345781.232  CMAC_LOG_RNG_REQ_QUEUED                     0
4d00h:  345781.272  CMAC_LOG_RNG_REQ_TRANSMITTED
4d00h:  345781.280  CMAC_LOG_RNG_RSP_MSG_RCVD
4d00h:  345781.282  CMAC_LOG_RNG_RSP_SID_ASSIGNED               3
4d00h:  345781.284  CMAC_LOG_ADJUST_RANGING_OFFSET              2288
4d00h:  345781.288  CMAC_LOG_RANGING_OFFSET_SET_TO              11898
4d00h:  345781.292  CMAC_LOG_ADJUST_TX_POWER                    7
4d00h:  345781.294  CMAC_LOG_POWER_LEVEL_IS                     24.0  dBmV (comma)
4d00h:  345781.298  CMAC_LOG_STATE_CHANGE                       ranging_2_state
4d00h:  345781.302  CMAC_LOG_RNG_REQ_QUEUED                     3
4d00h:  345782.298  CMAC_LOG_RNG_REQ_TRANSMITTED
4d00h:  345782.300  CMAC_LOG_RNG_RSP_MSG_RCVD
4d00h:  345782.304  CMAC_LOG_RANGING_SUCCESS
4d00h:  345782.316  CMAC_LOG_STATE_CHANGE                       dhcp_state
4d00h:  345782.450  CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS           10.1.1.25
4d00h:  345782.452  CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS           10.10.20.1
4d00h:  345782.456  CMAC_LOG_DHCP_TOD_SERVER_ADDRESS            10.10.20.2
4d00h:  345782.460  CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
4d00h:  345782.464  CMAC_LOG_DHCP_TZ_OFFSET                     0
```

```
4d00h:  345782.466  CMAC_LOG_DHCP_CONFIG_FILE_NAME              frequency.cm
4d00h:  345782.470  CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
4d00h:  345782.474  CMAC_LOG_DHCP_COMPLETE
4d00h:  345782.598  CMAC_LOG_STATE_CHANGE                       establish_tod_state
4d00h:  345782.606  CMAC_LOG_TOD_REQUEST_SENT
4d00h:  345782.620  CMAC_LOG_TOD_REPLY_RECEIVED                 3178880491
4d00h:  345782.628  CMAC_LOG_TOD_COMPLETE
4d00h:  345782.630  CMAC_LOG_STATE_CHANGE                       security_associate_state
4d00h:  345782.634  CMAC_LOG_SECURITY_BYPASSED
4d00h:  345782.636  CMAC_LOG_STATE_CHANGE                       configuration_file
4d00h:  345782.640  CMAC_LOG_LOADING_CONFIG_FILE                frequency.cm
4d00h: %LINEPROTO-5-UPDOWN:  Line protocol on Interface cable-modem0,changed state to up
4d00h:  345783.678  CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
NOTE—frequency override:
4d00h:  345783.682  CMAC_LOG_DS_FREQ_OVERRIDE                   535250000
4d00h:  345783.686  CMAC_LOG_STATE_CHANGE                       reset_hardware_state
4d00h:  345784.048  CMAC_LOG_STATE_CHANGE                       wait_for_link_up_state
4d00h:  345784.052  CMAC_LOG_DRIVER_INIT_IDB_RESET              0x082A5226
4d00h:  345784.054  CMAC_LOG_LINK_DOWN
4d00h:  345784.056  CMAC_LOG_LINK_UP
4d00h:  345784.062  CMAC_LOG_STATE_CHANGE                       ds_channel_scanning_state
4d00h:  345785.198  CMAC_LOG_DS_NO_QAM_FEC_LOCK                 535250000
4d00h:  345785.212  CMAC_LOG_DS_TUNER_KEEPALIVE
4d00h:  345787.018  CMAC_LOG_UCD_MSG_RCVD                       1
4d00h:  345787.022  CMAC_LOG_DS_64QAM_LOCK_ACQUIRED             453000000
```

Incorrect frequency specified in **cable modem change-frequency** on the Cisco uBR7246VXR can also cause the CM to switch frequencies, and if the frequency configured on the router is not chosen carefully, you will see results similar to that above. The **cable modem change-frequency** command on the Cisco uBR7246VXR is optional and is typically left out by default.

After a downstream channel has been acquired and latency has been calculated, the next task is to locate a suitable upstream channel. The cable modem listens for an upstream channel descriptor (UCD) which contains the physical properties of the upstream channel, such as frequency, modulation, channel width, and other parameters defined in the burst descriptors.

A cable modem that cannot find a usable upstream channel descriptor may be on a downstream channel for which no upstream service is provided. This is likely to be a faulty configuration of the router. You can check this using the **show controller cable *x*** command. Another possible reason a cable modem may not find a usable UCD is that its hardware or MAC may not support the parameters in the burst descriptors. This is likely to be either a faulty configuration of the Cisco uBR7246VXR or a CM that is not DOCSIS-compliant.

After a usable UCD is found, the cable modem begins to listen to MAP (Bandwidth Allocation Map) messages which contain the upstream bandwidth allocation map of time. A section of time is mapped out into mini-slots and assigned to individual modems. There are also regions in the MAP for broadcast, contention-based initial maintenance (or broadcast) ranging. It is these regions of the MAP that the cable modem must send its initial ranging requests until the Cisco uBR7246VXR responds with a ranging response (RNG-RSP).

If a CM cannot find an initial maintenance region before a T2 timer expires, check whether the Cisco uBR7246VXR has a faulty configuration. You should also check the **insertion-interval** for the cable interface on the router.

Ranging Process - init(r1),init(r2), and init(rc) state

At this stage, the CM begins a ranging process to calculate the necessary transmit power level to reach the Cisco uBR7246VXR at its desired input power level. A reasonably good transmit power is roughly 40 - 50 dBmV (based on a uBR7246 input power of 0 dBmV.) Other hardware may vary. Like the downstream channel, the carrier in the upstream channel should be sufficiently strong for the Cisco uBR7246VXR receiver to discern the symbols yet not too high to prevent increased bit error-rates.

The CM sends a ranging request (RNG-REQ) message to the Cisco uBR7246VXR and waits for a ranging response (RNG-RSP) message or a T3 timer expiry. If a T3 timeout occurs, the retry count increments. If the retry count is less than the maximum number of retries, the cable modem transmits another RNG-REQ at a higher power level. This ranging process occurs in the initial maintenance or broadcast regions of the MAP because the Cisco uBR7246VXR has not assigned the cable modem a service identifier (SID) for unicast transmissions in the MAP. Thus, broadcast ranging is contention based and subject to collisions. To compensate for this, the cable modems have a ranging backoff algorithm to calculate a random backoff time between RNG-REQ transmissions. This can be configured using **cable upstream range-backoff** command. When the transmit power reaches a sufficient level for the Cisco uBR7246VXR, it responds to the RNG-REQ with a RNG-RSP containing a temporary SID. This SID is used to identify unicast transmission regions in the MAP for unicast ranging.

The following output shows a CM in init(r1) state indicating the CM cannot get past the initial ranging stage:

```
#show cable modem
```

```
Interface Prim    Online     Timing   Rec     QoS   CPE   IP address   MAC address
Sid               State      Offset   Power

Cable2/0/U0 6     init(r1)   2813      12.00  2     0     10.1.1.22     0050.7366.1e01
```

The debug below shows how the CM fails to complete the ranging process and resets after a T3 timer expiry. Note the CMAC_LOG_**ADJUST_TX_POWER** messages coming from the Cisco uBR7246VXR asking the CM to adjust its power:

```
1w3d:      871160.618   CMAC_LOG_STATE_CHANGE                        ranging_1_state
1w3d:      871160.618   CMAC_LOG_RANGING_OFFSET_SET_TO               9610
1w3d:      871160.622   CMAC_LOG_POWER_LEVEL_IS                      19.0  dBmV (comman)
1w3d:      871160.622   CMAC_LOG_STARTING_RANGING
1w3d:      871160.622   CMAC_LOG_RANGING_BACKOFF_SET                 0
1w3d:      871160.622   CMAC_LOG_RNG_REQ_QUEUED                      0
1w3d:      871160.678   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      871160.682   CMAC_LOG_RNG_RSP_MSG_RCVD
1w3d:      871160.682   CMAC_LOG_RNG_RSP_SID_ASSIGNED                6
1w3d:      871160.682   CMAC_LOG_ADJUST_RANGING_OFFSET               2813
1w3d:      871160.682   CMAC_LOG_RANGING_OFFSET_SET_TO               12423
1w3d:      871160.686   CMAC_LOG_ADJUST_TX_POWER                     48
```

```
1w3d:      871160.686   CMAC_LOG_STATE_CHANGE                    ranging_2_state
1w3d:      871160.686   CMAC_LOG_RNG_REQ_QUEUED                  6
1w3d:      871161.690   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      871161.690   CMAC_LOG_RNG_RSP_MSG_RCVD
1w3d:      871161.694   CMAC_LOG_ADJUST_TX_POWER                 -36
1w3d:      871161.694   CMAC_LOG_RANGING_CONTINUE
1w3d:      871162.698   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      871162.898   CMAC_LOG_T3_TIMER
1w3d:      871163.734   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      871163.934   CMAC_LOG_T3_TIMER
1w3d:      871164.766   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      871164.966   CMAC_LOG_T3_TIMER
131.CABLEMODEM.CISCO: 1w3d: %UBR900-3-RESET_T3_RETRIES_EXHAUSTED: R03.0 Ranging
1w3d:      871164.966   CMAC_LOG_RESET_T3_RETRIES_EXHAUSTED
1w3d:      871164.966   CMAC_LOG_STATE_CHANGE                    reset_interface_state
1w3d:      871164.966   CMAC_LOG_STATE_CHANGE                    reset_hardware_state
Note: init(r1) is ranging_1_state and init(r2) is ranging_2_state
```

You can get an indication of the Transmit power on the CM by displaying the following command:

```
#sh cont cable-modem 0
BCM Cable interface 0:
CM unit 0, idb 0x2010AC, ds 0x86213E0, regaddr = 0x800000, reset_mask 0x80
station address 0050.7366.2223  default station address 0050.7366.2223
PLD VERSION: 32

MAC State is wait_for_link_up_state, Prev States = 2
MAC mcfilter 00000000  data mcfilter 00000000

MAC extended header ON
DS: BCM 3116 Receiver: Chip id = 2
US: BCM 3037 Transmitter: Chip id = 30AC

Tuner: status=0x00
Rx: tuner_freq 0, symbol_rate 5055932, local_freq 11520000
    snr_estimate 30640, ber_estimate 0, lock_threshold 26000
    QAM not in lock, FEC not in lock, qam_mode QAM_64
Tx: tx_freq 27984000, power_level 0x20 (8.0 dBmV), symbol_rate 8 (1280000 sym/s)
```

If a cable modem cannot proceed out of ranging state, the likely cause is an insufficient transmit power level. Transmit power can be adjusted by adjusting attenuation at the low frequency port. Increased attenuation will result in increased transmit power levels. 20 - 30 dBmV of attenuation is a good place to start.

After initial ranging init(r1) the cable modem proceeds to init(r2). This is where the cable modem must configure the transmit timing offset and power level to ensure that transmissions from the modem are received at the correct time and are at an acceptable input power level at the Cisco uBR7246VXR receiver. This is performed through a conversation of unicast RNG-REQ and RNG-RSP messages. The RNG-RSP messages contain the power and timing offset corrections that the cable modem must make.

The cable modem continues to transmit RNG-REQ and to perform adjustments per RNG-RSP until the RNG-RSP message indicates ranging success or ranging complete by reaching the init(rc) state. If a cable modem cannot proceed out of init (r2) the transmit power needs to be refined. Below is an output display of a CM in init(r2) state. Note the asterisk (*) symbol next to the Rec Power column indicating that the noise power adjustment method is active for this modem. An exclamation mark (!) means the cable modem has reached its maximum transmit power.

```
#show cable modem
```

```
Interface Prim    Online     Timing    Rec     QoS    CPE    IP address   MAC address
Sid               State      Offset    Power
Cable2/0/U0 5     init(r2)   2289      *4.00   2      0      10.1.1.25     0050.7366.2223
```

DHCP - init(d) state

The next stage after successful ranging is acquiring network configuration via DHCP.

Below is a an output display of **show cable modem** showing a cable modem in init(d), which indicates that the DHCP request was received from the CM:

```
#show cable modem
```

Interface Prim Sid	Online State	Timing Offset	Rec Power	QoS	CPE	IP address	MAC address
Cable2/0/U0 7	init(d)	2811	0.25	2	0	10.1.1.20	0030.96f9.65d9

Note that the CM can cycle through init(r1) to init(d) indefinitely. Following are some possible causes:

- IP connectivity issue from the Cisco uBR7246VXR to the DHCP server.
- DHCP server down.
- Wrong default gateway configured at the DHCP server.
- Low transmit power at the CM.

Note Although you can ping the DHCP server from the Cisco uBR7246VXR, the problem could be that the DHCP has the incorrect gateway set since it may be able to respond to the primary IP address of the Cisco uBR7246VXR cable interface, but not the secondary IP address, which is used as the source IP address during the DHCP discovery phase.

Frequently, errors at DHCP state manifest themselves as timeouts rather than NAKs. The order of DHCP messages should be DISCOVER, OFFER, REQUEST, ACK. If the cable modem is transmitting a DISCOVER with no OFFER response from the DHCP server, turn on UDP debugging on the Cisco uBR7246VXR using the following command:

```
# debug ip udp
```

```
4d01h:   UDP:    rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=584
4d01h:   BOOTP:  opcode 1 from host 0.0.0.0 on Cable2/0, 0 secs, 0 hops
4d01h:   UDP:    forwarded broadcast 67 from 10.1.1.10 to 10.20.20.9 on Ethernet0
4d01h:   UDP:    rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=584
4d01h:   BOOTP:  opcode 1 from host 0.0.0.0 on Cable2/0, 0 secs, 0 hops
4d01h:   UDP:    forwarded broadcast 67 from 10.1.1.10 to 10.20.20.9 on Ethernet0
```

```
4d01h:   UDP:    rcvd src=172.17.110.136(67), dst=10.1.1.10(67), length=314
4d01h:   BOOTP:  opcode 2 from host 10.20.20.9 on Ethernet1/0, 0 secs, 0 hops
4d01h:   BOOTP:  Broadcasting response 10.20.20.9 -> 10.1.1.20 (Cable2/0)
4d01h:   UDP:    forwarded broadcast 68 from 10.20.20.9 to 255.255.255.255 on Ca0
4d01h:   UDP:    rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=584
4d01h:   BOOTP:  opcode 1 from host 0.0.0.0 on Cable2/0, 0 secs, 0 hops
4d01h:   UDP:    forwarded broadcast 67 from 10.1.1.10 to 110.20.20.9 on Ethernet0
4d01h:   UDP:    rcvd src=10.20.20.9(67), dst=10.1.1.10(67), length=314
4d01h:   BOOTP:  opcode 2 from host 10.20.20.9 on Ethernet1/0, 0 secs, 0 hops
4d01h:   BOOTP:  Broadcasting response 10.20.20.9 -> 10.1.1.20 (Cable2/0)
4d01h:   UDP:    forwarded broadcunast 68 from 10.20.20.9 to 255.255.255.255 on 1
All possible debugging has been turned off
```

Caution Running debug commands on a uBR7246 (Universal Broadband Router) with more than a handful of modems may cause the Cisco uBR7246VXR to halt the system in order to keep up with the debugging. In this case, all the cable modems may lose sync and debugging will be useless.

If no packets are seen through debug messages, check the configuration of the **cable helper-address** statement on the cable interface to which this modem is attached. If this is configured correctly and a packet trace of the DHCP server subnet also reveals no DHCP packets from the cable modem, then look at the output errors of the cable modem's cable interface or the input errors of the cable interface of the Cisco uBR7246VXR. It might be a good idea to boost the transmitter power of the CM a bit more with more attenuation.

If packets are seen to be transmitted onto the DHCP server subnet, check the cable modem debug messages to see if there are parameter request or assignment errors. At this stage of troubleshooting you should investigate the routing between the CM and the DHCP server. Also, double-check the DHCP server configuration and the DHCP logs.

Below is a sample debug taken at the CM by running the **debug cable-modem mac log verbose** command:

```
1w3d:   865015.920   CMAC_LOG_RANGING_SUCCESS
1w3d:   865015.920   CMAC_LOG_STATE_CHANGE                        dhcp_state
1w3d:   865053.580   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:   865053.584   CMAC_LOG_RNG_RSP_MSG_RCVD
1w3d:   865055.924   CMAC_LOG_WATCHDOG_TIMER
131.CABLEMODEM.CISCO: 1w3d: %UBR900-3-RESET_DHCP_WATCHDOG_EXPIRED:
Cable Interface Reset due to DHCP watchdog timer expiration
1w3d:   865055.924   CMAC_LOG_RESET_DHCP_WATCHDOG_EXPIRED
1w3d:   865055.924   CMAC_LOG_STATE_CHANGE                        reset_interface_state
1w3d:   865055.924   cMAC_LOG_DHCP_PROCESS_KILLED
1w3d:   865055.924   CMAC_LOG_STATE_CHANGE                        reset_hardware_state
```

As you can see, the above the DHCP process failed and the cable modem was reset.

DHCP - init(i) state

After a reply to the DHCP request has been received and an IP address assigned to the cable modem the next **show cable modem** command gives its state as init(i):

```
#show cable modem


Interface Prim    Online     Timing    Rec     QoS    CPE    IP address   MAC address
Sid               State      Offset    Power

Cable2/0/U0 7     init(i)    2815      -0.25   2      0      10.1.1.20     0030.96f9.65d9
```

In the output above, the CM never gets beyond state init(i). Repetitive **show cable modem** displays will usually show the cable modem cycling between init(r1), init(r2), init(rc), init(d) and init(i) indefinitely.

Following is a list of the more common reasons for a cable modem not getting further than init(i):

- Incorrect or invalid DOCSIS file specified in the DHCP server.
- TFTP server issues, for example: incorrect IP address, TFTP server unreachable.
- Problems getting TOD or Timing Offset.
- Incorrect Router setting in the DHCP configuration.

Since the Cable Modem has reached as far as init(i) we know that it has got as far as obtaining an IP address. This is shown in the output display of the **debug cable-modem mac log verbose** command from the cable modem below:

```
3d20h:  334402.548   CMAC_LOG_RANGING_SUCCESS
3d20h:  334402.548   CMAC_LOG_STATE_CHANGE dhcp_state
NOTE—IP address Assigned to CM:
3d20h:  334415.492   CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS                 10.1.1.20
3d20h:  334415.492   CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS                 10.20.20.9
3d20h:  334415.492   CMAC_LOG_DHCP_TOD_SERVER_ADDRESS                  10.20.20.9
3d20h:  334415.492   CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
3d20h:  334415.492   CMAC_LOG_DHCP_TZ_OFFSET                           0
NOTE—DOCSIS file CM is trying to load:
3d20h:  334415.496   CMAC_LOG_DHCP_CONFIG_FILE_NAME                    nofile
3d20h:  334415.496   CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
3d20h:  334415.496   CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRESS
3d20h:  334415.496   CMAC_LOG_DHCP_COMPLETE
3d20h:  334415.508   CMAC_LOG_STATE_CHANGE                             establish_tod_state
3d20h:  334415.512   CMAC_LOG_TOD_REQUEST_SENT                         10.20.20.9
3d20h:  334415.524   CMAC_LOG_TOD_REPLY_RECEIVED                       3178343318
3d20h:  334415.524   CMAC_LOG_TOD_COMPLETE
3d20h:  334415.528   CMAC_LOG_STATE_CHANGE                             security_association
                                                                       _state
3d20h:  334415.528   CMAC_LOG_SECURITY_BYPASSED
3d20h:  334415.528   CMAC_LOG_STATE_CHANGE                             configuration_file
NOTE—DOCSIS file name:
```

```
3d20h:  334415.528    CMAC_LOG_LOADING_CONFIG_FILE                      nofile
133.CABLEMODEM.CISCO: 3d20h: %LINEPROTO-5-UPDOWN: Line protocol on Interface cap
3d20h:  334416.544    CMAC_LOG_CONFIG_FILE_TFTP_FAILED                  -1
3d20h:  334416.548    CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
3d20h:  334416.548    CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
```

Similarly, TFTP server issues give similar errors resulting in the cable modem resetting and cycling through the same process indefinitely:

```
3d21h:  336136.520    CMAC_LOG_STATE_CHANGE                     dhcp_state
3d21h:  336149.404    CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS          10.1.1.20
NOTE—TFTP Server address
3d21h:  336149.404    CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS         10.1.1.10
3d21h:  336149.404    CMAC_LOG_DHCP_TOD_SERVER_ADDRESS          10.1.1.20
3d21h:  336149.404    CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
3d21h:  336149.404    CMAC_LOG_DHCP_TZ_OFFSET                   0
3d21h:  336149.408    CMAC_LOG_DHCP_CONFIG_FILE_NAME            platinum.cm
3d21h:  336149.408    CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
3d21h:  336149.408    CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRES
3d21h:  336149.408    CMAC_LOG_DHCP_COMPLETE
3d21h:  336149.420    CMAC_LOG_STATE_CHANGE                     establish_tod_state
3d21h:  336149.424    CMAC_LOG_TOD_REQUEST_SENT                 10.1.1.20
3d21h:  336149.436    CMAC_LOG_TOD_REPLY_RECEIVED               3178345052
3d21h:  336149.436    CMAC_LOG_TOD_COMPLETE
3d21h:  336149.440    CMAC_LOG_STATE_CHANGE                     security_association_state
3d21h:  336149.440    CMAC_LOG_SECURITY_BYPASSED
3d21h:  336149.440    CMAC_LOG_STATE_CHANGE                     configuration_file
3d21h:  336149.440    CMAC_LOG_LOADING_CONFIG_FILE              platinum.cm
133.CABLEMODEM.CISCO: 3d21h: %LINEPROTO-5-UPDOWN: Line protocol on Interface cap
3d21h:  336163.252    CMAC_LOG_RNG_REQ_TRANSMITTED
3d21h:  336163.252    CMAC_LOG_RNG_RSP_MSG_RCVD
NOTE—TFTP process failing:
3d21h:  336165.448    CMAC_LOG_CONFIG_FILE_TFTP_FAILED          -1
3d21h:  336165.448    CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
3d21h:  336165.452    CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
3d21h:  336165.452    CMAC_LOG_STATE_CHANGE                     reset_interface_state
```

Problems getting TOD (Time of Day) or Timing Offset can also result in the cable modem not achieving online status:

```
3d21h:   338322.500    CMAC_LOG_STATE_CHANGE                       dhcp_state
3d21h:   338334.260    CMAC_LOG_RNG_REQ_TRANSMITTED
3d21h:   338334.260    CMAC_LOG_RNG_RSP_MSG_RCVD
```

```
3d21h:   338335.424    CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS             10.1.1.20
3d21h:   338335.424    CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS            10.10.10.1
3d21h:   338335.424    CMAC_LOG_DHCP_ERROR_ACQUIRING_TOD_ADDRESS
3d21h:   338335.424    CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
3d21h:   338335.424    CMAC_LOG_DHCP_ERROR_ACQUIRING_TZ_OFFSET
3d21h:   338335.424    CMAC_LOG_DHCP_CONFIG_FILE_NAME                platinum.cm
3d21h:   338335.428    CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
3d21h:   338335.428    CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRESS
3d21h:   338335.428    CMAC_LOG_DHCP_COMPLETE
3d21h:   338335.428    CMAC_LOG_RESET_DHCP_FAILED
3d21h:   338335.432    CMAC_LOG_STATE_CHANGE                        reset_interface_state
3d21h:   338335.432    CMAC_LOG_STATE_CHANGE                        reset_hardware_state
3d21h:   338336.016    CMAC_LOG_STATE_CHANGE                        wait_for_link_up_state
```

Note Prior to IOS version 12.1(1) TOD needed to be specified in the DHCP server in order for the cable modem to go online. However, after 12.1(1) TOD is not required but the cable modem still needs to get the timing offset, as shown below.

```
344374.528   CMAC_LOG_STATE_CHANGE                                 dhcp_state
344377.292   CMAC_LOG_RNG_REQ_TRANSMITTED
344377.292   CMAC_LOG_RNG_RSP_MSG_RCVD
344387.412   CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS                      10.1.1.20
344387.412   CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS                     10.10.10.1
NOTE— TOD server IP address obtained:
344387.412   CMAC_LOG_DHCP_TOD_SERVER_ADDRESS                      10.10.10.1
344387.412   CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
NOTE— Timing offset not specified in DHCP server:
344387.412   CMAC_LOG_DHCP_ERROR_ACQUIRING_TZ_OFFSET
344387.412   CMAC_LOG_DHCP_CONFIG_FILE_NAME                        platinum.cm
344387.412   CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
344387.412   CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRESS
344387.412   CMAC_LOG_DHCP_COMPLETE
344387.412   CMAC_LOG_RESET_DHCP_FAILED
NOTE—Modem resetting:
344387.412   CMAC_LOG_STATE_CHANGE                                 reset_interface_state
```

In the debug below there is no time-server specified, but there is a timing offset configured in the DHCP server, and therefore the cable modem goes online:

```
3d23h:    345297.516  CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS         10.1.1.20
3d23h:    345297.516  CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS         172.17.110.136
3d23h:    345297.516  CMAC_LOG_DHCP_ERROR_ACQUIRING_TOD_ADDRESS
```

```
3d23h:   345297.516  CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
3d23h:   345297.516  CMAC_LOG_DHCP_TZ_OFFSET                    0
3d23h:   345297.516  CMAC_LOG_DHCP_CONFIG_FILE_NAME             platinum.cm
3d23h:   345297.520  CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
3d23h:   345297.520  CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRESS
3d23h:   345297.520  CMAC_LOG_DHCP_COMPLETE
3d23h:   345297.532  CMAC_LOG_STATE_CHANGE                      establish_tod_state
3d23h:   345297.532  CMAC_LOG_TOD_NOT_REQUESTED_NO_TIME_ADDR
3d23h:   345297.532  CMAC_LOG_STATE_CHANGE                      security_association_state
3d23h:   345297.536  CMAC_LOG_SECURITY_BYPASSED
3d23h:   345297.536  CMAC_LOG_STATE_CHANGE                      configuration_file
3d23h:   345297.536  CMAC_LOG_LOADING_CONFIG_FILE               platinum.cm
3d23h:   345297.568  CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
3d23h:   345297.568  CMAC_LOG_STATE_CHANGE                      registration_state
3d23h:   345297.592  CMAC_LOG_REG_RSP_MSG_RCVD
3d23h:   345297.592  CMAC_LOG_COS_ASSIGNED_SID                  1/7
3d23h:   345297.596  CMAC_LOG_RNG_REQ_QUEUED                    7
3d23h:   345297.596  CMAC_LOG_REGISTRATION_OK
3d23h:   345297.596  CMAC_LOG_STATE_CHANGE                      establish_privacy_state
3d23h:   345297.596  CMAC_LOG_PRIVACY_NOT_CONFIGURED
3d23h:   345297.596  CMAC_LOG_STATE_CHANGE                      maintenance_state
133.CABLEMODEM.CISCO: 3d23h: %LINEPROTO-5-UPDOWN: Line protocol on Interface changed state
to up
```

Not including a Router option setting in the DHCP server or specifying an invalid IP address in the Router option field also results in a cable modem not getting beyond init(i) state, as shown in the output from **debug cable-modem mac log verbose** below:

```
1d16h:   146585.940   CMAC_LOG_CONFIG_FILE_TFTP_FAILED          -1
1d16h:   146585.940   CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
1d16h:   146585.944   CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
1d16h:   146585.944   CMAC_LOG_STATE_CHANGE                     reset_interface_state
1d16h:   146585.944   CMAC_LOG_STATE_CHANGE                     reset_hardware_state
```

TOD exchange- init(t) state

After a cable modem has acquired its network parameters, it must request the time of day from a Time Of Day (TOD) server. TOD uses a UTC timestamp (seconds from January 1, 1970) combined with the time offset option value from DHCP, to calculate the current time. The time is used for syslog and event log timestamps.

Below we have modems with SID 1 and 2 in init(t). Note that with IOS later than 12.1(1) the cable modem still comes online even though the TOD exchange failed, as you can see in the following output from the **show cable modem** command:

```
#show cable modem
```

```
Interface Prim     Online     Timing   Rec     QoS    CPE   IP address   MAC address
Sid                State      Offset   Power
Cable2/0/U0 1      init(t)    2808     0.00    2      0     10.1.1.20     0030.96f9.65d9
Cable2/0/U0 2      init(t)    2809     0.25    2      0     10.1.1.21     0030.96f9.6605
Cable2/0/U0 3      init(i)    2810     -0.25   2      0     10.1.1.22     0050.7366.1e01
```

```
2d01h:   177933.712   CMAC_LOG_STATE_CHANGE                      dhcp_state
2d01h:   177933.716   CMAC_LOG_RNG_REQ_TRANSMITTED
2d01h:   177933.716   CMAC_LOG_RNG_RSP_MSG_RCVD
2d01h:   177946.596   CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS          10.1.1.20
2d01h:   177946.596   CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS          10.20.20.1
2d01h:   177946.596   CMAC_LOG_DHCP_TOD_SERVER_ADDRESS           10.30.30.1
2d01h:   177946.596   CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
2d01h:   177946.596   CMAC_LOG_DHCP_TZ_OFFSET                    0
2d01h:   177946.600   CMAC_LOG_DHCP_CONFIG_FILE_NAME             platinum.cm
2d01h:   177946.600   CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
2d01h:   177946.600   CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRESS
2d01h:   177946.600   CMAC_LOG_DHCP_COMPLETE
2d01h:   177946.612   CMAC_LOG_STATE_CHANGE                      establish_tod_state
2d01h:   177946.716   CMAC_LOG_RNG_REQ_TRANSMITTED
2d01h:   177946.716   CMAC_LOG_RNG_RSP_MSG_RCVD
133.CABLEMODEM.CISCO: 2d01h: %LINEPROTO-5-UPDOWN: Line protocol on Interface cap
2d01h:   177947.716   CMAC_LOG_RNG_REQ_TRANSMITTED
2d01h:   177947.716   CMAC_LOG_RNG_RSP_MSG_RCVD
2d01h:   177948.616   CMAC_LOG_TOD_REQUEST_SENT                  10.30.30.1
2d01h:   177948.716   CMAC_LOG_RNG_REQ_TRANSMITTED
2d01h:   177954.616   CMAC_LOG_TOD_REQUEST_SENT                  10.30.30.1
2d01h:   177954.716   CMAC_LOG_RNG_REQ_TRANSMITTED
2d01h:   177954.716   CMAC_LOG_RNG_RSP_MSG_RCVD
2d01h:   177960.616   CMAC_LOG_TOD_REQUEST_SENT                  10.30.30.1
2d01h:   177960.712   CMAC_LOG_RNG_REQ_TRANSMITTED
2d01h:   177960.716   CMAC_LOG_RNG_RSP_MSG_RCVD
2d01h:   177961.716   CMAC_LOG_RNG_REQ_TRANSMITTED
131.CABLEMODEM.CISCO: 2d01h: %UBR900-3-TOD_FAILED_TIMER_EXPIRED:TOD failed, but Cable
Interface proceeding to operational state
2d01h:   177986.616   CMAC_LOG_TOD_WATCHDOG_EXPIRED
2d01h:   177986.616   CMAC_LOG_STATE_CHANGE                      security_association_state
2d01h:   177986.616   CMAC_LOG_SECURITY_BYPASSED
2d01h:   177986.616   CMAC_LOG_STATE_CHANGE                      configuration_file
2d01h:   177986.620   CMAC_LOG_LOADING_CONFIG_FILE               platinum.cm
2d01h:   177986.644   CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
2d01h:   177986.644   CMAC_LOG_STATE_CHANGE                      registration_state
```

```
2d01h:  177986.644  CMAC_LOG_REG_REQ_MSG_QUEUED
2d01h:  177986.648  CMAC_LOG_REG_REQ_TRANSMITTED
2d01h:  177986.652  CMAC_LOG_REG_RSP_MSG_RCVD
2d01h:  177986.652  CMAC_LOG_COS_ASSIGNED_SID             1/1
2d01h:  177986.656  CMAC_LOG_RNG_REQ_QUEUED               1
NOTE— Modem online:
2d01h:  177986.656  CMAC_LOG_REGISTRATION_OK
2d01h:  177986.656  CMAC_LOG_STATE_CHANGE                 establish_privacy_statee
2d01h:  177986.656  CMAC_LOG_PRIVACY_NOT_CONFIGURED
2d01h:  177986.656  CMAC_LOG_STATE_CHANGE                 maintenance_state
2d01h:  177988.716  CMAC_LOG_RNG_REQ_TRANSMITTED
```

Time of day errors almost always point to a DHCP misconfiguration. Possible misconfigurations that can result in TOD errors include the following:

- Gateway address misconfigurations.
- Incorrect TOD server address.

Make sure you can ping the time-server to rule out IP connectivity issues and to verify the time-server is available.

Option File Transfer Started-init(o) State

The main configuration and administration interface to the cable modem is the configuration file downloaded from the provisioning server. This configuration file contains downstream channel and upstream channel identification and characteristics as well as class of service settings, baseline privacy settings, general operational settings, network management information, software upgrade fields, filters, and vendor specific settings.

A cable modem stuck in init(o) state usually indicates that the cable modem started or is ready to start downloading the configuration file, but was unsuccessful. This can be due to the following reasons:

- Incorrect, corrupt, or missing DOCSIS configuration file.
- Unable to reach the TFTP server, because it is unavailable or there is no IP connectivity.
- Invalid or missing configuration parameters in DOCSIS file.
- Wrong file permissions on the TFTP server.

Note that you may not always see init(o), instead you might see init(i) and then cycling through from init(r1) to init(i). A more accurate state can be derived by displaying the output of **show controller cable-modem 0 mac stat**e.

Following is an abbreviated display of that output:

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

```
MAC State:              configuration_file_state
Ranging SID:            4
Registered:             FALSE
Privacy Established:    FALSE
```

When you see init(o) in the show cable modem output, running the **debug cable-modem mac log verbose** will not tell you if it is a corrupt configuration file or a TFTP server failure that is the cause. The debug points to both of them.

Examples of invalid configuration parameters in the Cisco Broadband Configurator (available for download on cisco.com) are invalid or missing Vendor ID or Vendor Specific Information. The result is similar to the following output display:

```
w3d:       880748.992   CMAC_LOG_STATE_CHANGE                        dhcp_state
1w3d:      880751.652   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      880751.656   CMAC_LOG_RNG_RSP_MSG_RCVD
1w3d:      880761.876   CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS            10.1.1.20
1w3d:      880761.876   CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS            10.10.10.1
1w3d:      880761.876   CMAC_LOG_DHCP_TOD_SERVER_ADDRESS             10.10.10.1
1w3d:      880761.876   CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
1w3d:      880761.876   CMAC_LOG_DHCP_TZ_OFFSET                      0
NOTE—Corrupt configuration file:
1w3d:      880761.880   CMAC_LOG_DHCP_CONFIG_FILE_NAME               data.cm
1w3d:      880761.880   CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
1w3d:      880761.880   CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRESS
1w3d:      880761.880   CMAC_LOG_DHCP_COMPLETE
1w3d:      880761.892   CMAC_LOG_STATE_CHANGE                        establish_tod_state
1w3d:      880761.896   CMAC_LOG_TOD_REQUEST_SENT                    10.10.10.1
1w3d:      880761.904   CMAC_LOG_TOD_REPLY_RECEIVED                  3180091733
1w3d:      880761.908   CMAC_LOG_TOD_COMPLETE
1w3d:      880761.908   CMAC_LOG_STATE_CHANGE                        security_association_state
1w3d:      880761.908   CMAC_LOG_SECURITY_BYPASSED
1w3d:      880761.912   CMAC_LOG_STATE_CHANGE                        configuration_file_state
1w3d:      880761.912   CMAC_LOG_LOADING_CONFIG_FILE                 data.cm
1w3d:      880762.652   CMAC_LOG_RNG_REQ_TRANSMITTED
1w3d:      880762.652   CMAC_LOG_RNG_RSP_MSG_RCVD
133.CABLEMODEM.CISCO: 00:13:07: %LINEPROTO-5-UPDOWN: Line protocol on Interface cable-modem0,
changed state to up
00:13:08: 788.004      CMAC_LOG_CONFIG_FILE_CISCO_BAD_TYPE          155
00:13:08: 788.004      CMAC_LOG_CONFIG_FILE_CISCO_BAD_TYPE          115
00:13:08: 788.004      CMAC_LOG_CONFIG_FILE_CISCO_BAD_TYPE          116
00:13:08: 788.004      CMAC_LOG_CONFIG_FILE_CISCO_BAD_ATTR_MAX_LENG 128
00:13:08: 788.008      CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
00:13:08: 788.008      CMAC_LOG_RESET_CONFIG_FILE_READ_FAILED
```

Online, Online(d), Online(pk), Online(pt) state

These states indicate that the cable modem has achieved online status and is able to transmit and receive data:

- Online
- Online(pk)
- Online(pt)

Online(d), however indicates that the cable modem has come online but has been denied network access.

```
#show cable modem
```

```
Interface Prim    Online        Timing   Rec      QoS    CPE    IP address   MAC address
Sid               State         Offset   Power
Cable2/0/U0 4     online        2810     -0.75    6      0      10.1.1.20     0030.96f9.65d9
Cable2/0/U0 5     online(pt)    2290     0.25     5      0      10.1.1.25     0050.7366.2223
Cable2/0/U0 6     online(d)     2815     0.00     6      0      10.1.1.27     0001.9659.4461
```

Online(d) is typically caused by disabling the Network Access option under Radio Frequency info in the Cisco Broadband Configurator (available for download on cisco.com). The default for Network Access is **enabled**. This can be seen in the display of **show cable modem** above and in the following output from **debug cable-modem mac log verbose**:

```
04:11:34: 15094.700    CMAC_LOG_STATE_CHANGE                       dhcp_state
04:11:46: 15106.392    CMAC_LOG_RNG_REQ_TRANSMITTED
04:11:46: 15106.396    CMAC_LOG_RNG_RSP_MSG_RCVD
04:11:47: 15107.620    CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS           10.1.1.20
04:11:47: 15107.620    CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS           172.17.110.136
04:11:47: 15107.620    CMAC_LOG_DHCP_TOD_SERVER_ADDRESS            172.17.110.136
04:11:47: 15107.620    CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
04:11:47: 15107.620    CMAC_LOG_DHCP_TZ_OFFSET                     0
NOTE—Network Access disabled:
04:11:47: 15107.624    CMAC_LOG_DHCP_CONFIG_FILE_NAME              noaccess.cm
04:11:47: 15107.624    CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
04:11:47: 15107.624    CMAC_LOG_DHCP_ERROR_ACQUIRING_LOG_ADDRES
04:11:47: 15107.624    CMAC_LOG_DHCP_COMPLETE
04:11:47: 15107.636    CMAC_LOG_STATE_CHANGE                       establish_tod_state
04:11:47: 15107.640    CMAC_LOG_TOD_REQUEST_SENT                   172.17.110.136
04:11:47: 15107.648    CMAC_LOG_TOD_REPLY_RECEIVED                 3179226080
04:11:47: 15107.652    CMAC_LOG_TOD_COMPLETE
04:11:47: 15107.652    CMAC_LOG_STATE_CHANGE                       security_association_state
04:11:47: 15107.652    CMAC_LOG_SECURITY_BYPASSED
04:11:47: 15107.652    CMAC_LOG_STATE_CHANGE                       configuration_file_state
04:11:47: 15107.652    CMAC_LOG_LOADING_CONFIG_FILE                noaccess.cm
```

```
133.CABLEMODEM.CISCO: 04:11:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface
cable-modem0, changed state to up
04:11:48: 15108.672    CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
04:11:48: 15108.672    CMAC_LOG_STATE_CHANGE                     registration_state
04:11:48: 15108.672    CMAC_LOG_REG_REQ_MSG_QUEUED
04:11:48: 15108.676    CMAC_LOG_REG_REQ_TRANSMITTED
04:11:48: 15108.680    CMAC_LOG_REG_RSP_MSG_RCVD
04:11:48: 15108.680    CMAC_LOG_COS_ASSIGNED_SID                 1/4
04:11:48: 15108.684    CMAC_LOG_RNG_REQ_QUEUED                   4
04:11:48: 15108.684    CMAC_LOG_NETWORK_ACCESS_DENIED
04:11:48: 15108.684    CMAC_LOG_REGISTRATION_OK
04:11:48: 15108.684    CMAC_LOG_STATE_CHANGE                     establish_privacy_state
04:11:48: 15108.684    CMAC_LOG_PRIVACY_NOT_CONFIGURED
04:11:48: 15108.684    CMAC_LOG_STATE_CHANGE                     maintenance_state
04:11:49: 15109.392    CMAC_LOG_RNG_REQ_TRANSMITTED
```

You can also check by examining the output of **show controllers cable-modem 0 mac state** on the cable modem:

Note The information in this example is for illustration and has been truncated.

```
Config File:
    Network Access: FALSE <<<<<<<<<<<<<<<<<<<<<< network access denied
Maximum CPEs:                     3
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:                4
    Max Downstream Rate:         10000000
    Max Upstream Rate:           1024000
    Upstream Priority:           7
    Min Upstream Rate:           0
    Max Upstream Burst:          0
    Privacy Enable:              FALSE
```

Online means the cable modem has come online and is able to communicate with the Cisco uBR7246VXR. If the baseline privacy interface (BPI) is not enabled, the online status is the default state (assuming the cable modem initialization was successful). If BPI is configured you will see a status of online(pk), followed shortly by online(pt). The following debug output display was taken on the cable modem side using the **debug cable-modem mac log verbose** command, showing only the registration part:

```
5d03h:    445197.804   CMAC_LOG_STATE_CHANGE                  registration_state
5d03h:    445197.804   CMAC_LOG_REG_REQ_MSG_QUEUED
5d03h:    445197.812   CMAC_LOG_REG_REQ_TRANSMITTED
5d03h:    445197.816   CMAC_LOG_REG_RSP_MSG_RCVD
5d03h:    445197.816   CMAC_LOG_COS_ASSIGNED_SID              1/4
5d03h:    445197.816   CMAC_LOG_RNG_REQ_QUEUED                4
5d03h:    445197.816   CMAC_LOG_REGISTRATION_OK
5d03h:    445197.816   CMAC_LOG_STATE_CHANGE                  establish_privacy_state
5d03h:    445197.820   CMAC_LOG_PRIVACY_FSM_STATE_CHANGE
machine: KEK, event/state: EVENT_1_PROVISIONED/STATE_A_START, new state:
STATE_B_AUTH_WAIT
5d03h:    445197.828   CMAC_LOG_BPKM_REQ_TRANSMITTED
5d03h:    445197.848   CMAC_LOG_BPKM_RSP_MSG_RCVD
5d03h:    445197.848   CMAC_LOG_PRIVACY_FSM_STATE_CHANGE
machine: KEK, event/state: EVENT_3_AUTH_REPLY/STATE_B_AUTH_WAIT, new state:
STATE_C_AUTHORIZED
5d03h:    445198.524   CMAC_LOG_PRIVACY_FSM_STATE_CHANGE
machine: TEK, event/state: EVENT_2_AUTHORIZED/STATE_A_START, new state: STATE_B_OP_WAIT
5d03h:    445198.536   CMAC_LOG_RNG_REQ_TRANSMITTED
5d03h:    445198.536   CMAC_LOG_RNG_RSP_MSG_RCVD
5d03h:    445198.536   CMAC_LOG_BPKM_REQ_TRANSMITTED
5d03h:    445198.536   CMAC_LOG_BPKM_RSP_MSG_RCVD
5d03h:    445198.540   CMAC_LOG_PRIVACY_FSM_STATE_CHANGE
machine: TEK, event/state: EVENT_8_KEY_REPLY/STATE_B_OP_WAIT, new state:
STATE_D_OPERATIONAL
5d03h:    445198.548   CMAC_LOG_PRIVACY_INSTALLED_KEY_FOR_SID 4
5d03h:    445198.548   CMAC_LOG_PRIVACY_ESTABLISHED
5d03h:    445198.552   CMAC_LOG_STATE_CHANGE                  maintenance_state
5d03h:    445201.484   CMAC_LOG_RNG_REQ_TRANSMITTED
5d03h:    445201.484   CMAC_LOG_RNG_RSP_MSG_RCVD
```

If there is a problem with BPI in general you will see reject(pk), meaning it could not get through the key authentication stage. This state is covered in the reject(pk) and reject (pt) section.

For correct BPI operation ensure that the Cisco uBR7246VXR and the cable modem are both running a BPI enabled image. This is indicated by the symbol **K1** in the image name.

Also, ensure that the field **Baseline Privacy Enable** is set to 1 under the Class of Service option in the Cisco Broadband Configurator (available for download on cisco.com). If the Cisco uBR7246VXR is running a BPI enabled image and the cable modem is not, and BPI is enabled in the Cisco Broadband Configurator, then you will observe the cable modem cycling between online and offline.

Reject(pk) and Reject(pt) state

The following output display from **show cable modem** on the Cisco uBR7246VXR shows a reject(pk) state:

```
#show cable modem


Interface Prim    Online        Timing    Rec     QoS    CPE   IP address   MAC address
Sid               State         Offset    Power

Cable2/0/U0 2     reject(pk)    2812      0.00    6      0     10.1.1.20     0030.96f9.65d9

01:58:51: %UBR7200-5-UNAUTHSIDTIMEOUT: CMTS deleted BPI unauthorized Cable Modem
0030.96f9.65d9
```

Often, when there is a problem with the BPI configuration you will see a reject(pk) state. This state is typically caused by the following:

- Corrupt public key by the CM in the auth request, refer to sample debug cable privacy for proper sequence of events.
- Presence of cable privacy authenticate-modem configuration command on the CMTS router but no Radius server present.
- Improperly configured Radius server.

Reject(pt) is typically caused by an invalid traffic encryption key (TEK). Following is the output display using **debug cable privacy** on the Cisco uBR7246VXR:

```
02:32:08: CMTS Received AUTH REQ.
02:32:08: Created a new CM key for 0030.96f9.65d9.
02:32:08: CMTS generated AUTH_KEY.
02:32:08: Input : 70D158F106B0B75
02:32:08: Public Key:
02:32:08: 0x0000: 30 68 02 61 00 DA BA 93 3C E5 41 7C 20 2C D1 87
02:32:08: 0x0010: 3B 93 56 E1 35 7A FC 5E B7 E1 72 BA E6 A7 71 91
02:32:08: 0x0020: F4 68 CB 86 A8 18 FB A9 B4 DD 5F 21 B3 6A BE CE
02:32:08: 0x0030: 6A BE E1 32 A8 67 9A 34 E2 33 4A A4 0F 8C DB BD
02:32:08: 0x0040: D0 BB DE 54 39 05 B0 E0 F7 19 29 20 8C F9 3A 69
02:32:08: 0x0050: E4 51 C6 89 FB 8A 8E C6 01 22 02 34 C5 1F 87 F6
02:32:08: 0x0060: A3 1C 7E 67 9B 02 03 01 00 01
02:32:08: RSA public Key subject:
02:32:08: 0x0000: 30 7C 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01 05
02:32:08: 0x0010: 00 03 6B 00 30 68 02 61 00 DA BA 93 3C E5 41 7C
02:32:08: 0x0020: 20 2C D1 87 3B 93 56 E1 35 7A FC 5E B7 E1 72 BA
02:32:08: 0x0030: E6 A7 71 91 F4 68 CB 86 A8 18 FB A9 B4 DD 5F 21
02:32:08: 0x0040: B3 6A BE CE 6A BE E1 32 A8 67 9A 34 E2 33 4A A4
02:32:08: 0x0050: 0F 8C DB BD D0 BB DE 54 39 05 B0 E0 F7 19 29 20
02:32:08: 0x0060: 8C F9 3A 69 E4 51 C6 89 FB 8A 8E C6 01 22 02 34
02:32:08: 0x0070: C5 1F 87 F6 A3 1C 7E 67 9B 02 03 01 00 01
02:32:08: RSA encryption result = 0
02:32:08: RSA encrypted output:
02:32:08: 0x0000: B6 CA 09 93 BF 2C 05 66 9D C5 AF 67 0F 64 2E 31
02:32:08: 0x0010: 67 E4 2A EA 82 3E F7 63 8F 01 73 10 14 4A 24 ED
02:32:08: 0x0020: 65 8F 59 D8 23 BC F3 A8 48 7D 1A 08 09 BF A3 A8
02:32:08: 0x0030: D6 D2 5B C4 A7 36 C4 A9 28 F0 6C 5D A1 3B 92 A2
02:32:08: 0x0040: BC 99 CC 1F C9 74 F9 FA 76 83 ED D5 26 B4 92 EE
02:32:08: 0x0050: DD EA 50 81 C6 29 43 4F 73 DA 56 C2 29 AF 05 53
02:32:08: CMTS sent AUTH response.
02:32:08: CMTS Received TEK REQ.
02:32:08: Created a new key for SID 2.
02:32:08: CMTS sent KEY response.
```

Below is a sample debug output on the cable modem when there is an authorization failure:

```
6d02h:    527617.480   CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE
6d02h:    527617.480   CMAC_LOG_STATE_CHANGE                    registration_state
6d02h:    527617.484   CMAC_LOG_REG_REQ_MSG_QUEUED
6d02h:    527617.488   CMAC_LOG_REG_REQ_TRANSMITTED
6d02h:    527617.492   CMAC_LOG_REG_RSP_MSG_RCVD
6d02h:    527617.492   CMAC_LOG_COS_ASSIGNED_SID                1/2
6d02h:    527617.492   CMAC_LOG_RNG_REQ_QUEUED                  2
6d02h:    527617.492   CMAC_LOG_REGISTRATION_OK
6d02h:    527617.496   CMAC_LOG_STATE_CHANGE                    establish_privacy_state
6d02h:    527617.496   CMAC_LOG_PRIVACY_FSM_STATE_CHANGE
machine: KEK, event/state: EVENT_1_PROVISIONED/STATE_A_START, new state:
STATE_B_AUTH_WAIT
6d02h:    527617.504   CMAC_LOG_BPKM_REQ_TRANSMITTED
6d02h:    527617.504   CMAC_LOG_BPKM_RSP_MSG_RCVD
6d02h:    527617.508   CMAC_LOG_PRIVACY_FSM_STATE_CHANGE
machine: KEK, event/state: EVENT_2_AUTH_REJECT/STATE_B_AUTH_WAIT, new state:
STATE_E_AUTH_REJ_WAIT
129.CABLEMODEM.CISCO: 6d02h: %CMBPKM-1-AUTHREJECT: Authorization request rejected by
CMTS: Unauthorized CM
6d02h:    527618.588   CMAC_LOG_RNG_REQ_TRANSMITTED
6d02h:    527618.592   CMAC_LOG_RNG_RSP_MSG_RCVD
```

Similarly, a **debug cable privacy** on the Cisco uBR7246VXR provides the following errors:

```
02:47:00: CMTS Received AUTH REQ.
02:47:00: Sending KEK REJECT.
02:47:05: %UBR7200-5-UNAUTHSIDTIMEOUT: CMTS deleted BPI unauthorized Cable Modem
0030.96f9.65d9
```

Note The cable modem keeps cycling from reject(pk) to init(r1) indefinitely.

Another error you might encounter is that due to encryption export restrictions, some vendor modems may require the following command on the Cisco uBR7246VXR in the interface configuration:

```
(config-if)#cable privacy 40-bit-des
```

Registration - reject (m) state

After configuration, the cable modem sends a registration request (REG-REQ) with a required subset of the configuration settings, and the cable modem and uBR7246 message integrity checks (MIC). The cable modem MIC is a hashed calculation over the configuration file settings, which provides a method for the cable modem to be sure the configuration file was not tampered with in transit. The Cisco uBR7246VXR MIC is similar except it also includes a setting for a cable shared-secret authentication string. This shared secret is known by the Cisco uBR7246VXR and the provisioning server, and ensures that only cable modems authenticated by provisioning servers will be allowed to register with the router.

A cable modem in reject(m) state has a bad Message Integrity Check (MIC), typically caused by:

- Mismatch between cable shared-secret under the cable interface and CMTS Authentication value under Miscellaneous option in the Cisco Broadband Configurator (downloadable from cisco.com).
- Corrupt configuration file (DOCSIS file).

To correct the problem, verify that you have a valid configuration file, and that the value under **CMTS Authentication** is identical to the value configured in **cable shared-secret <line>** under the cable interface. Also verify that the Cisco uBR7246VXR allows the creation of class of service profiles, or use a profile created by the Cisco uBR7246VXR.

Registration - reject (c) state

A cable modem that fails registration due to bad class of service (COS) has a state of reject(c). Typically this is caused by:

- uBR7246 is unable or unwilling to grant a particular requested COS.
- Misconfigured parameter(s) in Class of Service option in Cisco Broadband Configurator (available for download on cisco.com), for example, two classes of service with the same ID.

Following is the output from **debug cable-modem mac log verbose** taken on the cable modem side showing failure due to bad COS:

```
w3d:     885643.820  CMAC_LOG_STATE_CHANGE                 registration_state
1w3d:    885643.820  CMAC_LOG_REG_REQ_MSG_QUEUED
1w3d:    885643.824  CMAC_LOG_REG_REQ_TRANSMITTED
1w3d:    885643.828  CMAC_LOG_REG_RSP_MSG_RCVD
1w3d:    885643.828  CMAC_LOG_SERVICE_NOT_AVAILABLE        0x01,0x01,0x01
1w3d:    885643.828  CMAC_LOG_RESET_SERVICE_NOT_AVAILABLE
1w3d:    885643.828  CMAC_LOG_STATE_CHANGE                 reset_interface_state
1w3d:    885643.832  CMAC_LOG_STATE_CHANGE                 reset_hardware_state
1w3d:    885644.416  CMAC_LOG_STATE_CHANGE                 wait_for_link_up_state
1w3d:    885644.420  CMAC_LOG_DRIVER_INIT_IDB_RESET        0x8039E23C
1w3d:    885644.420  CMAC_LOG_LINK_DOWN
1w3d:    885644.420  CMAC_LOG_LINK_UP
1w3d:    885644.420  CMAC_LOG_STATE_CHANGE                 ds_channel_scanning_state
133.CABLEMODEM.CISCO:
1w3d:    %LINEPROTO-5-UPDOWN: Line protocol on Interface cable-modem0, changed state to
         down
1w3d:    885645.528  CMAC_LOG_UCD_MSG_RCVD                 1
1w3d:    885646.828  CMAC_LOG_DS_64QAM_LOCK_ACQUIRED       453000000
```

Similarly, **debug cable registration** on the Cisco uBR7246VXR gives the following output:

```
sniper#debug cable registration
CMTS registration debugging is on
sniper#
1d04h: %UBR7200-5-CLASSFAIL: Registration failed for Cable Modem 0001.9659.4461 on
interface Cable2/0/U0:
Bad/Missing Class of Service Config in REG-REQ
```

Note how the cable modem eventually resets and starts all over again.