Data-Over-Cable Service Interface Specifications
DOCSIS 1.1

Operations Support System Interface Specification

CM-SP-OSSIv1.1-C01-050907

CLOSED SPECIFICATION

Document Status Sheet

Document Control Number:	CM-SP-OSSIv1.1-C01-050907			
Revision History:	I01 – First Interim Release, April 7, 2000 I02 – Second Interim Release, July 14, 2000 I03 – Third Interim Release, December 15, 2000 I03 – Re-posting of I03, December 20, 2000 I04 – Fourth Interim Release, August 29, 2001 I05 – Fifth Issued Release, March 1, 2002 I06 – Sixth Issued Release, August 30, 2002 I07 – Seventh Issued Release, July 30, 2003 C01 – Closed September 7, 2005			
Date:	September 7, 2005			
Status Code:	~~Work in Process~~	~~Draft~~	~~Issued~~	Closed
Distribution Restrictions:	~~CableLabs & Members Only~~	~~CableLabs, Members, and Vendors Only~~		Public

Key to Document Status Codes

Work in Process	An incomplete document, designed to guide discussion and generate feedback, that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks:

Table of Contents

Figures

Tables

This page left blank intentionally.

1 Scope and Purpose

1.1 Scope

This Specification defines the Network Management requirements for support a DOCSIS® 1.1 environment. More specifically, the specification details the SNMP v3 protocol and how it coexists with SNMP V1/V2. The RFCs and Management Information Base (MIB) requirements are detailed as well as interface numbering, filtering, event notifications, etc. Basic network management principals such as account, configuration, fault, and performance management are incorporated in this specification for better understanding of managing a high-speed cable modem environment.

1.2 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This work or the adjective "REQUIRED" means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word of the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighted before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace required it or because it enhances the product, for example; another vendor may omit the same item.

This document defines many features and parameters and a valid range for each parameter is usually specified. Equipment (CM and CMTS) requirements are always explicitly stated. Equipment must comply with all mandatory (MUST and MUST NOT) requirements to be considered compliant with this specification. Support of non-mandatory features and parameter values is optional.

2 SNMP Protocol

The SNMPV3 protocol has been selected as the communication protocol for management of data-over-cable Services and MUST be implemented. Although SNMPv3 offers advantages, many management systems may not be capable of supporting SNMPV3 agents; therefore, support of SNMPv1 and SNMPv2c is also required and MUST be implemented.

The following IETF SNMP related RFCs MUST be implemented:

RFC-3410	Introduction and Applicability Statements for Internet Standard Management Framework
RFC-3411	An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks
RFC-3412	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
RFC-3413	Simple Network Management Protocol (SNMP) Applications
RFC-3414	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
RFC-3415	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
RFC-3416	Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)
RFC-3417	Transport Mappings for the Simple Network Management Protocol (SNMP)
RFC-3418	Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)
RFC-2576	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
RFC-1901	Introduction to Community-based SNMPv2
RFC-1157	A Simple Network Management Protocol

For support of SMIv2 the following IETF SNMP related RFCs MUST be implemented:

RFC-2578	Structure of Management Information Version 2 (SMIv2)
RFC-2579	Textual Conventions for SMIv2
RFC-2580	Conformance Statements for SMIv2

For support of Diffie-Helman Key exchange for the User Based Security Model, the follow IETF SNMP related RFC MUST be implemented:

RFC-2786	Diffie-Helman USM Key Management Information Base and Textual Convention

2.1 SNMP Mode for DOCSIS 1.1 compliant CMTS[1]

DOCSIS 1.1 compliant CMTS MUST support SNMPv1, SNMPv2c, and SNMPv3 and SNMP coexistence as described by RFC-3411-RFC-2576 and MAY support SNMPv1, SNMPv2c vendor proprietary solutions, including SNMP v1/v2c NmAccess mode, with the following requirements:

a) DOCSIS 1.1 compliant CMTS MUST operate in SNMP coexistence mode (not using docsDevNmAccessTable); additionally, SNMP coexistence mode MAY be disabled, by vendor

[1] Omnibus changes in Section 2.1 per ECN OSS-N-03066 by GO on 07/10/03.

proprietary configuration control, to allow the CMTS to support SNMPv1, SNMPv2c vendor proprietary solutions, including SNMP v1/v2c NmAccess mode (using docsDevNmAccessTable).

b) CMTS in SNMPv1/v2c NmAccess mode (using DOCS-CABLE-DEVICE-MIB docsDevNmAccess Table) MUST operate with the following requirements/limitations:
   - Only SNMPv1/v2c packets are processed
   - SNMPv3 pack3ets are dropped
   - docsDevNmAccessTable controls SNMP access and SNMP trap destinations as described in RFC-2669
   - None of the SNMPv3 MIBs as defined in [RFC-3411-3415] and [RFC-2576] are accessible.[2]

c) CMTS SNMPv1, SNMPv2c vendor proprietary solutions MUST operate with the following requirements/limitations:
   - Only SNMPv1/v2c packets are processed
   - SNMPv3 packets are dropped
   - Vendor proprietary solution MUST control SNMP access and SNMP trap destinations
   - None of the SNMPv3 MIBs as defined in [RFC-3411-3415] and [RFC-2576] are accessible.[3]

d) CMTS SNMP Coexistence Mode MUST operate with the following requirements/limitations:
   - SNMP v1/v2c/v3 Packets are processed as described by RFC-3411-3414 and RFC-2576.
   - docsDevNmAccessTable is not accessible. (If the CMTS also support DOCS-CABLE-DEVICE-MIB)
   - Access control and trap destinations are determined by the SNMP-COMMUNITY-MIB, SNMP-NOTIFICATION-MIB, SNMP-TARGET-MIB, SNMP-VIEW-BASED-ACM-MIB, SNMP-COMMUNITY-MIB, and SNMP-USER-BASED-SM-MIB.
   - The SNMP-COMMUNITY-MIB controls the translation of SNMPv1/v2c packet community string into securityName which select entries in the SNMP-USER-BASED-SM-MIB. Access control is provided by the SNMP-VIEW-BASED-ACM-MIB.
   - The SNMP-USER-BASED-SM-MIB and SNMP-VIEW-BASED-ACM-MIB control SNMPv3 packets.
   - Trap destinations are specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB.

2.1.1 Key Change Mechanism

DOCSIS 1.1 compliant CMTS SHOULD use the key-change mechanism specified in the RFC-2786. CMTS MUST always support the key-change mechanism described in the RFC-3414[4] to comply with industry-wide SNMP V3 standard.

2.2 SNMP Mode for DOCSIS 1.1 compliant CMs[5]

DOCSIS 1.1 compliant CMs (in 1.1 and 1.0 mode) MUST support SNMPv1, SNMPv2c and SNMPv3 as well as SNMP-coexistence (RFC-2576) with the following requirements:

a) Before completion of registration, the CM MUST operate as follows (in some CCCM implementations, SNMP access MAY be made inaccessible from the CPE for security reasons; in

[2] Revised this bullet statement per ECN OSS-N-03013 and 03066 by GO on 02/25/03 and 07/10/03.

[3] Revised this bullet statement per ECN OSS-N-03013 and 03066 by GO on 02/25/03 and 07/10/03.

[4] Revised RFC per ECN OSS-N-03066 by GO on 07/10/03.

[5] Ominbus changes to Section 2.2 per ECN OSS-N-03066 by GO on 07/10/03.

such implementation, the access to similar set of MIB objects SHOULD be provided by a diagnostic utility as described in section 6.3):

- IP connectivity between the CM and the SNMP management station MUST implemented as described in section 6.1
- The CM MUST provide read-only access to the following MIB objects:
  docsIfDownChannelFrequency
  docsIfDownChannelPower
  docsIfCmStatusValue
  docsDevServerBootState
  docsDevEventTable[6]
- The CM MAY provide read-only access to the following MIB objects
  sysDescr
  sysUptime
  ifTable
  ifXtable
  docsIfUpChannelFrequency
  docsIfSigQSignalQualityTable
  docsIfCmCmtsAddress
  docsIfCmStatusTxPower
  docsDevSwCurrentVers
- The CM MAY provide access to additional information, but MUST NOT reveal:
  CoS and QoS service flow information
  configuration file contents
  Secure Software Download information
  Key authentication and encryption material
  SNMP management and control
  DOCSIS functional modules statistics and configuration
  Network provisioning hosts and servers IPs addresses.
- Access from the RF interface MUST NOT be allowed
- SNMPv1/v2c packets are accepted which contain any community string
- All SNMPv3 packets are dropped
- The registration request MUST be sent and registration MUST be completed after successful processing of all MIB elements in the config file, but before beginning the calculation of the public values in the USMDHKickstart Table.

b) The content of the CM config file determines the CM SNMP mode after registration

- CM is in SNMPv1/v2c docsDevNmAccess Mode if the CM configuration file contains ONLY docsDevNmAccessTable setting for SNMP access control.
- If configuration file does not contain SNMP access control items (docsDevNmAccessTable or snmpCommunityTable or TLV 34.1/34.2 orTLV38), then the CM is in NmAccess mode.
- CM is in SNMP coexistence mode if the CM configuration file contains
  - snmpCommunityTable setting and/or
  - TLV type 34.1 and 34.2. and/or
  - TLV type 38
  - In this case, any entries made to the docsDevNmAccessTable are ignored.

c) After completion of registration - Modem operates in one of 2 modes. The operating mode is determined by the contents of the config file as described above.

[6] Added statement to this bullet list per ECN OSS-N-02192, chg #4, by GO, on 12/06/02.

SNMP V1/V2c NmAccess Mode (using docsDevNmAccess Table)

- Only SNMP V1/V2c packets are processed
- SNMP V3 packets are dropped
- docsDevNmAccessTable controls access and trap destinations as described in RFC-2669
- None of the SNMP V3 MIBs as defined in [RFC-3411-3415] and [RFC-2576] are accessible.[7]

SNMP Coexistence Mode

During calculation of USMDHKickstartTable public values:

- The modem MUST NOT allow any SNMP access from the RF port
- The modem MAY continue to allow access from the CPE port with the limited access as configured by the SNMP-COMMUNITY-MIB, SNMP-TARGET-MIB, SNMP-VIEW-BASED-ACM-MIB and SNMP-USER-BASED-SM-MIB.

After calculation of USMDHKickstartTable public values:

- The modem MUST send the cold start or warm start trap to indicate that the modem is now fully SNMPv3 manageable.
- SNMP V1/V2c/V3 Packets are processed as described by RFC-3411-3415 and RFC-2576.
- docsDevNmAccessTable is not accessible.
- Access control and trap destinations are determined by the SNMP-COMMUNITY-MIB, SNMP-NOTIFICATION-MIB, SNMP-TARGET-MIB, SNMP-VIEW-BASED-ACM-MIB, SNMP-COMMUNITY-MIB and SNMP-USER-BASED-SM-MIB.
- The SNMP-COMMUNITY-MIB controls the translation of SNMPv1/v2c packet community string into security name which select entries in the SNMP-USER-BASED-SM-MIB. Access control is provided by the SNMP-VIEW-BASED-ACM-MIB.
- SNMP-USER-BASED-SM-MIB and SNMP-VIEW-BASED-ACM-MIB controls SNMPv3 packets.
- Trap destinations are specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB.

d) In case of failure to complete SNMPv3 initialization (i.e. NMS can not access CM via SNMPv3 PDU), the CM is in the co-existence mode and will allow SNMPv1/v2c access if and only if the SNMP-COMMUNITY-MIB entries (and related entries) are configured.

2.2.1 SNMPv3 Initialization and Key changes [8]

DOCSIS 1.1 compliant CM MUST support the “SNMPv3 Initialization” and “DH Key Changes” requirements specified in the following sections.

The DOCSIS 1.1 cable modem is designated as having "very-secure" security posture in the context of RFC-3414 Appendix A and RFC-3415 Appendix A. This means that default usmUser and vacmAccess entries defined in RFC-3414 Appendix A and RFC-3415 Appendix A MUST NOT be present.

2.2.2 SNMPv3 Initialization

1. For each of up to 5 different security names, the Manager generates a pair of numbers:

   a. Manager generates a random number Rm

[7] Revised this bullet statement per ECN OSS-N-03013 by GO on 02/25/03.

[8] Revised RFC references per ECN OSS-N-03066 by GO on 07/10/03.

b. Manager uses DH equation to translate Rm to a public number z

z = g ^ Rm MOD p where g is the from the set of Diffie-Hellman parameters, p is the prime from those parameters

2. CM configuration file is created to include (security name, public number) pair and CM MUST support a minimum of 5 pairs. For example:

TLV type 34.1 (SnmpV3 Kickstart Security Name) = docsisManager

TLV type 34.2 (SnmpV3 Kickstart Public Number) = z

CM MUST support VACM entries defined in section 2.2.4 “VACM Profile”.

During the CM boot up process, the above values (security name, public number) will (MUST) be populated in the usmDHKickstartTable.

At this point:

usmDHKickstartMgrpublic.1 = “z” (octet string)
usmDHKickstartSecurityName.1 = “docsisManager”

When usmDHKickstartMgrpublic.n is set with a valid value during the registration, a corresponding row is created in the usmUserTable with the following values:

usmUserEngineID: localEngineID
usmUserName: usmDHKickstartSecurityName.n value
usmuserSecurityName: usmDHKickstartSecurityName.n value
usmUserCloneForm: ZeroDotZero
usmUserAuthProtocol: usmHMACMD5AuthProtocol
usmuserAuthKeyChange: derived from set value
usmUserOwnAuthKeyChange: derived from set value
usmUserPrivProtocol: usmDESPrivProtocol
usmUserPrivKeyChange: derived from set value
usmUserOwnPrivKeyChange: derived from set value
usmUserPublic: ""
usmUserStorageType: permanent
usmUserStatus: active

Note: For (CM) dhKickstart entries in usmUserTable, Permanent means it MUST be written to but not deleted and is not saved across reboots.

After the CM has registered with the CMTS.

1. CM generates a random number xa for each row populated in the usmDHKickstartTable which has a non zero length usmDHKickstartSecurityName and usmDHKickstartMgrPublic.

2. CM uses DH equation to translate xa to a public number c (for each row identified above)

c = g ^ xa MOD p where g is the from the set of Diffie-Helman parameters, p is the prime from those parameters

At this point:

usmDHKickstartMyPublic.1 = “c” (octet string)
usmDHKickstartMgrPublic.1 = “z” (octet string)
usmDHKickstartSecurityName.1 = “docsisManager”

3. CM calculate shared secret sk where sk = z ^xa mod p

4. CM uses sk to derive the privacy key and authentication key for each row in usmDHKickstartTable and sets the values into the usmUserTable

   As specified in RFC-2786, the privacy key and the authentication key for the associated username, "docsisManager" in this case, is derived from sk by applying the key derivation function PBKDF2 defined in PKCS#5v2.0.

```
privacy key <---     PBKDF2( salt =  0xd1310ba6,
                         iterationCount = 500,
                         keyLength = 16,
                         prf = id-hmacWithSHA1)
authentication  key  <----  PBKDF2( salt = 0x98dfb5ac,
                              iterationCount = 500,
                              keyLength = 16 (usmHMACMD5AuthProtocol),
                              prf = id-hmacWithSHA1)
```

   At this point the CM has completed its SNMPv3 initialization process and MUST allow appropriate access level to a valid securityName with the correct authentication key and/or privacy key.

   DOCSIS 1.1 compliant CM MUST properly populate keys to appropriate tables as specified by the SNMPv3 related RFCs and RFC-2786.

5. The following describes the process that the manager uses to derive CM's unique authentication key and privacy key.

   The SNMP manager accesses the contents of the usmDHKickstartTable using the security name of 'dhKickstart' with no authentication.

   DOCSIS 1.1 compliant CM MUST provide preinstalled entries in the USM table and VACM tables to correctly create user 'dhKickstart' of security level noAuthnoPriv that has read only access to system group and usmDHkickstartTable.

   SNMP manager gets the value of CM's usmDHKickstartMypublic number associated with the security name that manager wants to derive authentication and privacy keys for. With the manager's knowledge of the private random number, the manager can calculate the DH shared secret. From that shared secret, the manager can derive operational authentication and confidentiality keys for the security name that the manager is going to use to communicate with the CM.

2.2.3 DH Key Changes

DOCSIS 1.1 compliant CM MUST support the key-change mechanism specified in the RFC-2786.

2.2.4 VACM Profile

This section will address the default VACM profile for DOCSIS CM when it is operating in SNMP Coexistence mode.

The following VACM entries MUST be included by default in a compliant CM:

- The system manager, with full read/write/config access

  vacmSecurityModel: 3 (USM)
  vacmSecurityName:
  docsisManager
  vacmGroupName: docsisManager

vacmSecurityToGroupStorageType: permanent
vacmSecurityToGroupStatus: active

- An operator/CSR with read/reset access to full modem

  vacmSecurityModel: 3 (USM)
  RF Monitoring with read access to RF plant statistics
  vacmSecurityModel: 3 (USM)
  vacmSecurityName: docsisMonitor
  vacmGroupName: docsisMonitor
  vacmSecurityToGroupStorageType: permanent
  vacmSecurityToGroupStatus: active

- User debugging with read access to "useful" variables

  vacmSecurityModel: 3 (USM)
  vacmSecurityName: docsisUser
  vacmGroupName: docsisUser
  vacmSecurityToGroupStorageType: permanent
  vacmSecurityToGroupStatus: active

- Group name to view translations

  vacmGroupName: docsisManager
  vacmAccessContextPrefix:
  vacmAccessSecurityModel: 3 (USM)
  vacmAccessSecurityLevel: AuthPriv
  vacmAccessContextMatch: exact
  vacmAccessReadViewName: docsisManagerView
  vacmAccessWriteViewName: docsisManagerView
  vacmAccessNotifyViewName: docsisManagerView
  vacmAccessStorageType: permanent
  vacmAccessStatus: active

  vacmGroupName: docsisOperator
  vacmAccessContextPrefix:
  vacmAccessSecurityModel: 3 (USM)
  vacmAccessSecurityLevel: AuthPriv & AuthNoPriv
  vacmAccessContextMatch: exact
  vacmAccessReadViewName: docsisManagerView
  vacmAccessWriteViewName: docsisOperatorWriteView
  vacmAccessNotifyViewName: docsisManagerView
  vacmAccessStorageType: permanent
  vacmAccessStatus: active

  vacmGroupName: docsisMonitor
  vacmAccessContextPrefix:
  vacmAccessSecurityModel: 3 (USM)
  vacmAccessSecurityLevel: AuthNoPriv
  vacmAccessContextMatch: exact
  vacmAccessReadViewName: docsisMonitorView
  vacmAccessWriteViewName:
  vacmAccessNotifyViewName: docsisMonitorView
  vacmAccessStorageType: permanent
  vacmAccessStatus: active

  vacmGroupName: docsisUser
  vacmAccessContextPrefix:
  vacmAccessSecurityModel: 3 (USM)

vacmAccessSecurityLevel: AuthNoPriv
vacmAccessContextMatch: exact
vacmAccessReadViewName: docsisUserView
vacmAccessWriteViewName:
vacmAccessNotifyViewName:
vacmAccessStorageType: permanent
vacmAccessStatus: active

- The views

  docsisManagerView

  subtree: 1.3.6.1 (Entire mib).

  docsisOperatorWriteView

  subtree: docsDevBase
  subtree: docsDevSoftware
  subtree: docsDevEvControl
  subtree: docsDevEvThrottleAdminStatus

  docsisMonitorView

  subtree: 1.3.6.1.2.1.1 (system)
  subtree: docsIfBaseObjects
  subtree: docsIfCmObjects

  docsisUserView

  subtree: 1.3.6.1.2.1.1 (system)
  subtree: docsDevBase
  subtree: docsDevSwOperStatus
  subtree: docsDevSwCurrentVersion
  subtree: docsDevServerConfigFile
  subtree: docsDevEventTable
  subtree: docsDevCpeTable
  subtree: docsIfUpstreamChannelTable
  subtree: docsIfDownstreamChannelTable
  subtree: docsIfSignalQualityTable
  subtree: docsIfCmStatusTable

  DOCSIS 1.1 compliant CM MUST also support additional VACM users as they are configured via an SNMP-embedded configuration file.

3 Management Information Bases (MIBs)

This section defines the minimum set of managed objects required to support the management of CM and CMTS. Vendors MAY augment this MIB with objects from other standard or vendor-specific MIBs where appropriate.

DOCSIS OSSI 1.1 specification has priority over IETF MIB specification. Vendor MUST implement MIB requirements in accordance with the texts specified in OSSI 1.1 specification. Certain objects are deprecated or obsolete but may be required by the OSSI specification as mandatory and MUST be implemented.

Deprecated objects are optional. That is, a vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object MUST be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

Optional object. A vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object MUST be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

Obsolete object. It is optional. A vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object MUST be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

Section 3.1 and 3.2 include an overview of the MIB modules required for the management of the facilities specified in SP-RFI-1.1 and BPI+ specifications.

3.1 IPCDN Drafts and Others

Table 1. IPCDN Drafts[9, 10, 11]

REFERENCE	MIB	Applicable Device(s)
[IETF4]	IETF Proposed Standard RFC-version of Qos MIB, **"draft-ietf-ipcdn-qos-mib-04.txt"** DOCS-QOS-MIB	CM and CMTS
[IETF6]	IETF Proposed Standard RFC-version of BPI+ MIB, "**draft-ietf-ipcdn-bpiplus-mib-05.txt"** DOCS-BPI2-MIB	CM and CMTS
[IETF7]	IETF Proposed Standard RFC-version of USB MIB, "**dolnik-usb-mib-00.txt"** USB-MIB	CM only
[IETF9]	IETF Proposed Standard RFC-version of Subscriber Management MIB, "**draft-ietf-ipcdn-subscriber-mib-02.txt"** DOCS-SUBMGT-MIB	CMTS only
[IETF11]	IETF Proposed Standard RFC-version of RF MIB, "**draft-ietf-ipcdn-docs-rfmibv2-05.txt"** DOCS-IF-MIB	CM and CMTS

[9] Second row, changed text from "07" to "05", per ECN OSS-N-03020 (rescinds OSS-N-02229), by GO on 03/21/03.

[10] Fifth row, changed text from "04" to "05 per OSS-N-03022 by GO on 03/20/03.

[11] Revised Table 1 per ECN OSS-N-03066 by GO on 07/10/03.

3.2 IETF RFCs

Table 2. IETF RFCs[12]

REFERENCE	MIB	Applicable Device(s)
[RFC-2669]	DOCSIS Cable Device MIB: DOCS-CABLE-DEVICE-MIB	CM and CMTS
[RFC-3083]	Baseline Privacy Interface MIB: DOCS-BPI-MIB	CM
[RFC-2933]	Internet Group Management Protocol MIB: IGMP-STD-MIB	CM and CMTS
[RFC-2863]	The Interfaces Group MIB using SMIv2: IF-MIB	CM and CMTS
[RFC-2665]	Ethernet Interface MIB: EtherLike-MIB	CM and CMTS
[RFC-1493]	Bridge MIB: BRIDGE-MIB	CM and CMTS
[RFC-2011]	SNMPv2 Management Information Base for the Internet Protocol using SMIv2: IP-MIB	CM and CMTS
[RFC-2013]	Management Information Base for the User Datagram Protocol using SMIv2: UDP-MIB	CM and CMTS
[RFC-3418]	Management Information Base (MIB) for the Simple Network Management Protocol (SNMP): SNMPv2-MIB	CM and CMTS
[RFC-3410] [RFC-3411] [RFC-3412] [RFC-3413] [RFC-3414] [RFC-3415] [RFC-2576]	SNMP v3 MIBs: SNMP-FRAMEWORK-MIB, SNMP-MPD-MIB, SNMP-NOTIFICATION-MIB, SNMP-TARGET-MIB, SNMP-USER-BASED-SM-MIB, SNMP-VIEW-BASED-ACM-MIB, SNMP-COMMUNITY-MIB	CM and CMTS
[RFC-2786]	RFC-2786: Diffie-Helman USM Key: SNMP-USM-DH-OBJECTS-MIB	CM and CMTS

3.3 Managed Objects Requirements

The following sections detail any additional implementation requirements for the RFCs listed. Reference Appendix A for specific object implementation requirements.

The CM and CMTS MUST support a minimum of 10 available SNMP Table Rows unless otherwise specified by RFC or DOCSIS specification. The CM/CMTS minimum number of available SNMP Table Rows SHOULD mean rows (per table) that are available to support device configuration. CM/CMTS used (default) SNMP Table Row entries MUST NOT apply to the minimum number of available SNMP Table Rows.

3.3.1 CMTS MIB requirements

DOCSIS 1.1 compliant CMTS MUST implement Subscribe Management MIB.

[12] Table updated per ECN OSS-N-03066 by GO on 07/10/03.

3.3.2 Requirements for RFC-2669

RFC-2669 MUST be implemented by DOCSIS 1.1 compliant CMs. DOCSIS 1.1 compliant CMTS MUST implement mandatory required objects (as specified by Appendix A), and SHOULD implement the other non-mandatory required objects.

3.3.3 Requirements for DOCS-IF-MIB

The DOCS-IF-MIB MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

The docsIfDownChannelPower object-type MUST be implemented in a CMTS that provides an integrated RF upconverter. If the CMTS relies on an external upconverter, then the CMTS SHOULD implement the docsIfDownChannelPower object-type. The CMTS transmit power reported in the MIB object MUST be within 2 dB of the actual transmit power in dBmV when implemented. IF transmit power management is not implemented, the MIB object will be read-only and report the value of 0 (zero).

The docsIfDownChannelPower object-type MUST be implemented in DOCSIS 1.1 conforming CM's. This object is read-only. When operated at nominal line voltage, at normal room temperature, the reported power MUST be within 3 dB of the actual received channel power. Across the input power range from -15 dBmV to +15 dBmV, for any 1 dB change in input power, the CM MUST report a power change in the same direction that is not less than 0.5 dB. and not more than 1.5 dB.

The access of docsIfDownChannelFrequency object MUST be implemented as RW if a CMTS is in control of the downstream frequency. But if a CMTS provides IF output, docsIfDownChannelFrequency MUST be implemented as read-only and return 0.

All objects added as a result of the DOCS-IF-MIB [13]upgrade from RFC2670 to draft-ietf-ipcdn-docs-rfmibv2-05.txt are optional for DOCSIS 1.1 devices, with the exception of objects 'transferred' from the docsIfExt MIB, and objects indicating the CM modulation type. These objects are mandatory for DOCSIS 1.1 devices, and include docsIfDocsisBaseCapability, docsIfCmStatusDocsisOperMode, docsIfCmStatusModulationType, docsIfCmtsCmStatusDocsisRegMode, and docsIfCmtsCmStatusModulationType. docsIfCmtsChannelUtilizationTable, docsIfCmtsDownChannelCounterTable, and only the first nine MIB objects of docsIfCmtsUpChannelCounterTable**.** Refer to Appendix A for details on optional/mandatory status of new DOCS-IF-MIB [14]objects.[15]

“The docsIfQosProfMaxTransmitBurst range MUST be the same as the one defined in the RFIv1.1 specification, section C.1.1.4.6 “Maximum Upstream Channel Transmit Burst Configuration Setting” which has range 0 to 65535.”[16]

If the CMTS implements the docsIfUpChannelStatus object-type, the CMTS MUST NOT allow it to be set from active(1) directly or indirectly to destroy(6). The CMTS MUST return a wrongValue error. Entries with docsIfUpChannelStatus set to active(1) are logically linked to a physical interface, not temporarily created to clone parameters.[17]

---

[13] Revised requirement per ECN OSS-N-03066 by GO on 07/10/03.

[14] Revised requirement per ECN OSS-N-03066 by GO on 07/10/03.

[15] Revised “04” to “05” and added text per OSS-N-03022 by GO on 03/20/03.

[16] Added paragraph to Section 3.3.3 per ECN OSS-N-02219, by GO, on 12/02/02.

[17] Last paragraph of Section 3.3.3 added per OSS-N-02141 by RKV on 10/24/02.

3.3.4 Requirements for RFC-2863[18]

RFC-2863 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

The CMTS/CM ifAdminStatus object MUST provide administrative control over both MAC interfaces and individual channel and MUST be implemented as RW.

The ifType object has been assigned the following enumerated values for each instance of a Data Over Cable Service (DOCS) interface:

CATV MAC interface:: docsCableMacLayer (127)
CATV downstream channel: docsCableDownstream (128)
CATV upstream channel:: docsCableUpStream (129)

3.3.4.1 Interface Organization and Numbering

Assigned interface numbers for CATV-MAC and Ethernet (Ethernet-like interface) are used in both the NMAccessTable and IP/LLC filtering table to configure access and traffic policy at these interfaces. These configurations are generally encoded in the configuration file using TLV encoding. To avoid provisioning complexity the interface-numbering scheme MUST comply with the following requirements:

An instance of IfEntry MUST exist for each CATV-MAC interface, downstream channel, upstream channel, and each LAN interface enabled by the CM. The enablements of LAN interfaces MAY be fixed a prior during manufacturing process or MAY be determined dynamically during operation by the CM according to if an interface has a CPE device attached to it or not.[19]

If the CM has multiple CPE interfaces but only one CPE interface can be enabled at any given time, then the ifTable MUST only contain the entry corresponding to the enabled or the default CPE interface. If a MAC interface consists of more than one upstream and downstream channel, then a separate instance of ifEntry MUST also exist for each channel.

The ifStack group ([RFC-2863]) must be implemented to identify relationship among sub-interfaces. Note that the CATV-MAC interface MUST exist, even though it is broken out into sub-interfaces.

The example below illustrates a MAC interface with one downstream and two upstream channels for a CMTS.

[18] RFC # updated to 2863 throughout document per OSS-N-02190 by GO on 11/15/02.

[19] Revised paragraph per ECN OSS-N-03070 by GO on 07/11/03.

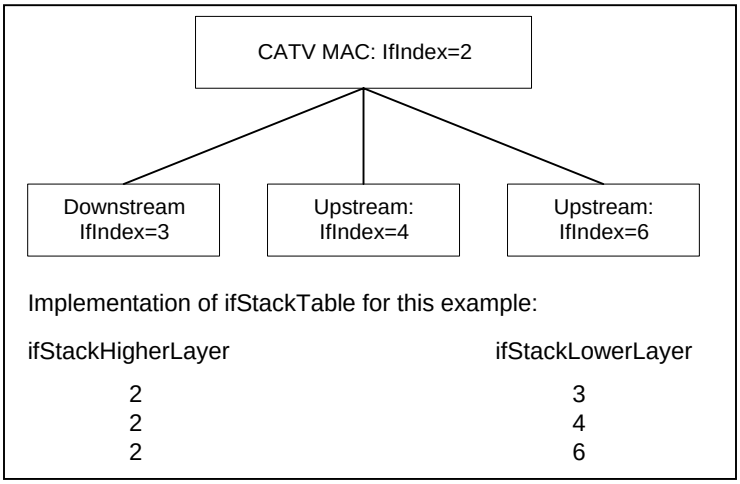

Figure 1. Ifindex Example for CMTS

At the CMTS, interface number is at the discretion of the vendor, and SHOULD correspond to the physical arrangement of connections. If table entries exist separately for upstream and downstream channels, then the ifStack group ([RFC-2863]) MUST be implemented to identify the relationship among sub-interfaces. Note that the CATV MAC interface(s) MUST exist, even if further broken out into sub-interfaces.

At the CM, interface MUST be numbered as:

Table 3. CM Interface numbering[20]

Interfaces	Type
1	primary CPE interface
2	CATV-MAC
3	RF-down
4	RF-Up
5 – 15, 32+n	Other interfaces
16 - 31	Other interfaces (Reserved)

If CM has more than one CPE interface, then the vendor MUST define which of (n) CPE interfaces is the primary CPE interface. The definition of the primary CPE interface MAY be fixed a prior during manufacturing process or MAY be determined dynamically during operation by the CM according to which interface has a CPE device attached to it. Regardless how many CPE interfaces the CM has or how the primary CPE interface is defined, the primary interface MUST be interface number 1.

The definition of the secondary CPE interface MAY be fixed a prior during manufacturing process or MAY be determined dynamically during operation by the CM according to which interface has a CPE device attached to it. The secondary CPE, and other interfaces, will start at 5.

DOCSIS CM may have multiple interfaces. If filter(s) (Ip, LLC, or NmAccess) are applied to CM IfIndex 1, the same filter(s) MUST also be applied to the "Other interfaces" (IfIndexes 5 and above); however, filters are never used to limit traffic between the CPE and “Other” interfaces within the CM.[21]

[20] Table 3 updated per ECN OSS-N-02213, chg #1, by GO, on 12/02/02.

[21] Replaced sentence per ECN OSS-N-02213, chg #2, by GO, on 12/02/02.

3.3.4.2 docsIfCmStatusValue and ifOperStatus Relationship

For CM RF downstream, RF upstream and RF MAC interfaces; the following are the expected relationship of ifOperStatus and docsIfCmStatusValue when ifAdminStatus = up (taken from DOCS-IF-MIB).

Table 4. docsIfCmStatusValue and ifOperStatus Relationship[22]

IfOperStatus	docsIfCmStatusValue
down(2):	other(1), notReady(2)
dormant(5):	notSynchronized(3), phySynchronized(4), usParametersAcquired(5), rangingComplete(6), ipComplet(7), todEstablished(8), paramTransferComplete(10), accessDenied(13)
up(1):	registrationComplete(11), securityEstablished(9), operational(12)

3.3.4.2.1 ifOperStatus and traffic

If the CM and CMTS interface's ifAdminStatus = down, the interface MUST not accept or forward any traffic (traffic includes data and MAC management traffic).

3.3.5 Interface MIB and Trap Enable

Interface MIB and Trap Enable specified in RFC-2863 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

If a multi-layer interface model is present in the device, each sub-layer for which there is an entry in the ifTable can generate linkUp/Down traps. Since interface state changes would tend to propagate through the interface stack (from top to bottom, or bottom to top), it is likely that several traps would be generated for each linkUp/Down occurrence. The CM and CMTS MUST implement the ifLinkUpDownTrapEnable object to allow managers to control trap generation, and configure only the interface sub-layers of interest.

The default setting of ifLinkUpDownTrapEnable MUST limit the number of traps generated to one, per interface, per linkUp/Down event. Interface state changes, of most interest to network managers, occur at the lowest level of an interface stack.

On CM linkUp/Down event a trap SHOULD be generated by the CM MAC interface and not by any sub-layers of the interface. Therefore, the default setting of ifLinkUpDownTrapEnable for CM MAC MUST be set to enable, and the default setting of ifLinkUpDownTrapEnable for CM RF-Up MUST be set to disable, and the default setting of ifLinkUpDownTrapEnable for CM RF-Down MUST be set to disable.

On CMTS interfaces (MAC, RF-Downstream(s), RF-Upstream(s)) the linkUp/Down event/trap SHOULD be generated by each CMTS interface. Therefore, the default setting of ifLinkUpDownTrapEnable for each CMTS interface (MAC, RF-Downstream(s), RF-Upstream(s)) MUST be set to enable.

3.3.6 Requirements for RFC-2665

RFC-2665 MUST be implemented by DOCSIS 1.1 compliant CMTS and CM if Ethernet or Fast Ethernet interfaces are present.

[22] Table 4 updated per ECN OSS-N-03064 by GO on 07/10/03.

3.3.7 Requirements for RFC-1493

RFC-1493 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

In both the CM and the CMTS (if the CMTS implements transparent bridging), the Bridge MIB ([RFC-1493]) MUST be implemented to manage the bridging process.

In the CMTS that implements transparent bridging, the Bridge MIB MUST be used to represent information about the MAC Forwarder states.

3.3.8 Requirements for RFC-2011

RFC-2011 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

3.3.8.1 The IP Group

The IP group MUST be implemented. It does not apply to IP packets forwarded by the device as a link-layer bridge. For the CM, it applies only to the device as an IP host. At the CMTS, it applies to the device as an IP host, and as a routers if IP routing is implemented.

3.3.8.2 The ICMP Group

The ICMP group MUST be implemented. It does not apply to IP packets forwarded by the device as a link-layer bridge. For the CM, it applies only to the device as an IP host. At the CMTS, it applies to the device as an IP host, and as a routers if IP routing is implemented.

Since CMs do not generate ICMP requests and do not support ICMP Timestamps, Table 24 lists MIB objects that are optional.

3.3.9 Requirements for RFC-2013

RFC-2013 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs. The UDP group does not apply to IP packets forwarded by the device as a link-layer bridge. For the CM, it applied only to the device an IP host. At the CMTS, it applies to the device only as an IP host.

3.3.10 Requirements for RFC-3418[23]

RFC-3418 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

3.3.10.1 The System Group

The System Group from RFC-3418 MUST be implemented. [24]

[23] Revised Section 3.3.10 per ECN OSS-N-03066 by GO on 07/10/03.

[24] Revised Section 3.3.10 per ECN OSS-N-03066 by GO on 07/10/03.

3.3.10.2 The SNMP Group

The SNMP Group from RFC-3418 MUST be implemented. [25]

3.3.11 Requirements for DOCS-QOS-MIB[26]

"draft-ietf-ipcdn-qos-mib-04.txt" MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs

The default values for the MIB objects in docsQosParamSetTable and docsQosServiceClassTable MUST follow the referenced ones in the RFIv1.1 specification. For example, docsQosParamSetMaxTrafficBurst default value is 3044 (which is 1522 * 2), docsQosServiceClassMaxTrafficBurst DEFVAL is 3044, docsQosParamSetMaxConcatBurst default value is 1522, and docsQosServiceClassMaxConcatBurst DEFVAL is 1522. If in the future, there are any related default values changed in the RFIv1.1 specification, the related default values in DOCS-QOS-MIB docsQosParamSetTable and docsQosServiceClassTable MUST be changed accordingly even though the MIB file is not changed in time.[27]

3.3.12 Requirements for "draft-ietf-ipcdn-igmp-mib-01.txt"

"draft-ietf-ipcdn-igmp-mib-01.txt" requirements have been deleted for CMTS and CMs.

3.3.13 Requirements for RFC-2933

RFC-2933 MUST be implemented by DOCSIS 1.1 compliant CMTS and CMs.

Refer to "Appendix J, "Application of RFC-2933 to DOCSIS 1.1 active/passive IGMP devices" for DOCSIS 1.1 IGMP cable device implementation details.

3.3.14 Requirements for DOCS-BPI2-MIB[28]

"draft-ietf-ipcdn-bpiplus-mib-~~07~~05.txt" MUST be implemented by DOCSIS 1.1 compliant CMTS and CM as specified in Appendix A.[29]

3.3.15 Requirements for USB-MIB[30]

(Note: Until the USB-MIB becomes an IETF RFC, the draft text will be available on the DOCSIS website.)

---

[25] Revised Section 3.3.10 per ECN OSS-N-03066 by GO on 07/10/03.

[26] Revised Section 3.3.11 per ECN OSS-N-03066 by GO on 07/10/03.

[27] Added paragraph to the section per ECN OSS-N-02214, by GO on 11/21/02.

[28] Revised Section 3.3.14 per ECN OSS-N-03066 by GO on 07/10/03.

[29] Changed "07" to "05" per ECN OSS-N-03020 (rescinds OSS-N-02229), by GO on 03/21/03.

[30] Revised Section 3.3.15 per ECN OSS-N-03066 by GO on 07/10/03.

3.3.16 Requirements for DOCS-SUBMGT-MIB[31]

"draft-ietf-ipcdn-subscriber-mib-02-.txt" MUST be implemented by DOCSIS 1.1 compliant CMTS.

DOCSIS 1.1 compliant CMTS MUST support a minimum number of filter groups; (30) thirty groups of (20) twenty filters each.

3.3.17 Requirements for RFC-2786[32]

RFC-2786 MUST be implemented by DOCSIS 1.1 compliant CMs. It (RFC-2786) MAY be implemented on the CMTS.

3.3.18 Requirements for RFC-3083[33]

RFC-3083 MUST be implemented by DOCSIS 1.1 compliant CMs as specified in Appendix A.

Due to the editorial error in RFC-3083, the DOCSIS 1.1 compliant CM MUST use the following definition for docsBpiCmAuthState and not the definition in RFC-3083.

```
docsBpiCmAuthState      OBJECT-TYPE
SYNTAX    INTEGER {
            start(1),
            authWait(2),
            authorized(3),
            reauthWait(4),
            authRejectWait(5)
      }
MAX-ACCESS  read-only

STATUS      current

DESCRIPTION
"The value of this object is the state of the CM authorization FSM. The start
state indicates that FSM is in its initial state."

REFERENCE
"DOCSIS Baseline Privacy Interface Specification, Section 4.1.2.1."
::= {docsBpiCmBaseEntry 3 }
```

In addition, compliant CMs MAY create new entries in the docsBpiCmTEKTable for any multicast SID(s) it receives in Auth-Reply messages. If implemented, the multicast SID MUST be used as an index in the docsBpiCmTEKTable in the docsIfCmServiceId field. Note that if the multicast SID is used in the docsBpiCmTEKTable, there MUST NOT be a corresponding entry in the docsIfCmServiceTable for the multicast SID, due to the definition of the docsIfCmService ID in the DOCS-IF-MIB.[34]

[31] Revised Section 3.3.16 per ECN OSS-N-03066 by GO on 07/10/03.

[32] Revised Section 3.3.17 per ECN OSS-N-03066 by GO on 07/10/03.

[33] Revised Section 3.3.18 per ECN OSS-N-03066 by GO on 07/10/03.

[34] New text added per OSS-N-02198 by GO on 11/21/02

3.3.19 Requirement for DOCS-IF-EXT-MIB

A DOCSIS 1.1 compliant CM/CMTS MAY support the DOCS-IF-EXT MIB, which is defined in Appendix K. If a DOCSIS 1.1 CM/CMTS supports the deprecated docsIfExt MIB objects in the docsCableDevice MIB trap definitions, then it MUST also support the DOCS-IF-EXT MIB.

3.3.20 Requirements for DOCS-CABLE-DEVICE-TRAP-MIB

DOCSIS 1.1 compliant CM/CMTS must implement DOCS-CABLE-DEVICE-TRAP-MIB, as specified in Appendix M.

3.3.21 Requirements for SNMPv3 MIBs

DOCSIS 1.1 compliant CM/CMTS MUST implement the MIBs defined in RFC 3411-3415 and RFC 2576.[35]

For CMs, the default value for any SNMPv3 object with a storageType textual convention MUST be 'volatile'. This overrides the default value specified in RFC 3413-3415 and RFC 2576. [36]

The CM MUST only accept the value of 'volatile' on any SNMPv3 storageType object.

An attempted set to a value of other(1), nonVolatile(3), permanent(4), or readOnly(5) will result an ‘inconsistentValue’ error. Values other than the valid range (1-5) would result a ‘wrongValue’ error.

The CM and CMTS SHOULD support a minimum of 30 available rows in the vacmViewTreeFamilyTable object.

3.4 CM Configuration Files, TLV-11 and MIB OIDs/Values

The following sections define the use of CM configuration file TLV-11 elements and the CM rules for translating TLV-11 elements into SNMP PDU (SNMP MIB OID/instance and MIB OID/instance value combinations; also referred to as SNMP varbinds).

This section also defines the CM behaviors, or state transitions, after either pass or fail of the CM configuration process.

For TLV-11 definitions refer to [DOCSIS 5; Appendix C].

3.4.1 CM configuration file TLV-11 element translation (to SNMP PDU)

TLV-11 translation defines the process used by CM to convert CM configuration file information (TLV-11 elements) into SNMP PDU (varbinds). The CM MUST translating CM configuration file TLV-11 elements into a single SNMP PDU containing (n) MIB OID/instance and value components (SNMP varbinds). Once a single SNMP PDU is constructed, the CM will process the SNMP PDU and determine CM configuration pass/fail based on the rules for CM configuration file processing, described below. However, if a CM is not physically capable of processing a, potentially large, single CM configuration file generated SNMP PDU,

[35] Revised RFC references per ECN OSS-N-03066 by GO on 07/10/03.

[36] Revised RFC references per ECN OSS-N-03066 by GO on 07/10/03.

then the CM must still behave as if all MIB OID/instance and value components (SNMP varbinds), from CM configuration file TLV-11 elements, are processed as a single SNMP PDU.

In accordance with [RFC-3416[37]], the single CM configuration file generated SNMP PDU will be treated "as if simultaneous" and the CM must behave consistently, regardless of the order in which TLV-11 elements appear in the CM configuration file, or SNMP PDU. The singular CM configuration file generated SNMP PDU requirement is consistent with SNMP PDU packet behaviors, received from an SNMP manager; SNMP PDU varbind order does not matter, and there is no defined MAX SNMP PDU limit.

The CM configuration file MUST NOT contain duplicate TLV-11 elements (duplicate means SNMP MIB object has either identical OID or OID from the old and new MIB that actually point to the same SNMP MIB object). If duplicate TLV-11 elements are received by the CM, from the CM configuration file, then the CM MUST fail CM configuration.

3.4.1.1 Rules for CreateAndGo and CreateAndWait

The CM MUST support CreateAndGo for row creation.

The CM MAY support CreateAndWait; with the constraint that CM configuration file TLV-11 elements MUST NOT be duplicated (all SNMP MIB OID/instance must be unique). For instance, an SNMP PDU, constructed from CM configuration file TLV-11 elements, which contains an SNMP CreateAndWait value, for a given SNMP MIB OID/instance, MUST NOT also contain an SNMP Active value for the same SNMP MIB OID/instance (and vice versa). A CM configuration file MAY contain a TLV-11 CreateAndWait element if the intended result is to create an SNMP table row which will remain in the SNMP NotReady or SNMP NotInService state until a non-configuration file SNMP PDU is issued, from an SNMP manager, to update the SNMP table row status.

Both SNMP NotReady and SNMP NotInService states are valid table row states after an SNMP CreateAndWait instruction.

3.4.2 Ignore CM configuration TLV-11 elements which are not supported by CM

If any CM configuration file TLV-11 elements translate to SNMP MIB OIDs that are not MIB OID elements supported by the CM, then those SNMP varbinds MUST be ignored, and treated as if they had not been present, for the purpose of CM configuration. This means that the CM will ignore SNMP MIB OIDs for other vendor's private MIBs as well as standard MIB elements that the CM does not support.

CMs that do not support SNMP CreateAndWait for a given SNMP MIB table MUST ignore, and treated as if not present, the set of columns associated with the SNMP table row.

If any CM configuration file TLV-11 element(s) are ignored, then the CM MUST report via the CM configured notification mechanism(s), after the CM is registered. The CM notification method MUST be in accordance with the "Standard DOCSIS event" section, defined within this document.

3.4.3 CM state after CM configuration file processing success

After successful CM configuration, via CM configuration file, CM MUST proceed to register, with CMTS, and pass data.

[37] Revised RFC reference per ECN OSS-N-03066 by GO on 07/10/03.

3.4.4 CM state after CM configuration file processing failure

If any CM configuration file generated SNMP PDU varbind performs an illegal set operation (illegal, bad, or inconsistent value) to any MIB OID/instance supported by the CM, then processing of the CM configuration file MUST fail. Any CM configuration file generated SNMP PDU varbind set failure MUST cause a CM configuration failure, and the CM MUST NOT proceed with CM registration.

3.5 Treatment and Interpretation of MIB Counters on the CM

Octet and packet counters implemented as counter32 and counter64 MIB objects are defined to be monotonically increasing positive integers with no specific initial value and a maximum value based on the counter size that will roll-over to zero when it is exceeded. In particular, counters are defined such that the only meaningful value is the difference between counter values as seen over a sequence of counter polls. However there are two situations that can cause this consistent monotonically increasing behavior to change: 1) resetting the counter due to a system or interface reinitialization or 2) a rollover of the counter when it reaches its maximum value of 2**32-1 or 2**64-1. In these situations, it must be clear what the expected behavior of the counters should be.

Case 1: Whenever the state of an interface changes resulting in an “interface counter discontinuity” as defined in RFC-2863. In this case the value of the ifXTable.ifXEntry.ifCounterDiscontinuityTime for the affected interface MUST be set to the current value of sysUpTime and ALL counters for the affected interface MUST be set to ZERO. Setting the ifAdminStatus of specified interface to down(2) MUST NOT be considered as an interface reset.

Case 2: SNMP Agent Reset. In this case, the value of the sysUpTime MUST be set to ZERO, all interface ifCounterDiscontinuityTime values MUST be set to ZERO, and all interface counters MUST be set to ZERO. Also, all other counters being maintained by the SNMP Agent MUST be set to ZERO.

Case 3: Counter Rollover. When a counter32 object reaches its maximum value of 4,294,967,295 the next value MUST be ZERO. When a counter64 object reaches its maximum value of 18,446,744,073,709,551,615 the next value MUST be ZERO. Note that unless a CM or CMTS vendor provides a means outside of SNMP to preset a counter64 or counter32 object to an arbitrary value, it will not be possible to test any rollover scenarios for counter64 objects (and many counter32 objects as well). This is because it is not possible for these counters to rollover during the service life of the device (see discussion in RFC-2863 section 3.1.6).

3.6 Config File Element – SNMP V3Notification Receiver[38]

The following sections detail the CM Configuration File TLV-38 “DOCSIS V3 Notification Receiver” mapping into SNMP V3 functional tables. A CM MUST support a minimum of 10 TLV-38 elements in a configuration file. For TLV-38 definitions refer to [DOCSIS 5; Appendix C].

Upon receiving one TLV 38, the CM MUST make entries to the following tables in order to cause the desired trap transmission: snmpNotifyTable, snmpTargetAddrTable, snmpTargetAddrExtTable, snmpTargetParamsTable, snmpNotifyFilterProfileTable, snmpNotifyFilterTable, nmpCommunityTable, usmUserTable, vacmSecurityToGroupTable, vacmAccessTable, and vacmViewTreeFamilyTable

A config file MAY also contain TLV MIB elements that make entries to any of the 10 tables listed above. These TLV MIB elements MUST NOT use index columns that start with the characters "@config".

[38] Revised Section 3.6 per ECN OSS-N-03066 by GO on 07/10/03.

3.6.1 Mapping of TLV fields into created SNMP V3 Table rows

The tables in this section show how the fields from the Config file TLV element (the tags in angle brackets <>) are placed into the SNMP V3 tables.

The correspondence between TLV fields and table tags <TAG> is shown below:

<IP Address> TLV 38.1
<Port> - TLV 38.2
<Trap type> TLV 38.3
<Timeout> TLV 38.4
<Retries> TLV 38.5
<Filter OID> TLV 38.6
<Security Name> TLV 38.7

These tables are shown in the order that the agent will search down through them when a notification is generated in order to determine who to send the notification to and how to fill out the contents of the notification packet.

3.6.1.1 snmpNotifyTable

Create 2 rows with fixed values, if 1 or more TLV elements are present

Table 5. snmpNotifyTable

snmpNotifyTable (RFC-2573 - SNMP-NOTIFICATION-MIB	1st Row	2nd Row
Column Name (* = Part of Index)	Column Value	Column Value
* snmpNotifyName	"@config_inform"	"@config_trap"
snmpNotifyTag	"@config_inform"	"@config_ trap "
snmpNotifyType	inform (2)	trap (1)
snmpNotifyStorageType	volatile	volatile
snmpNotifyRowStatus	Active (1)	Active (1)

3.6.1.2 snmpTargetAddrTable

Create 1 row for each TLV element in the config file

Table 6. snmpTargetAddrTable

snmpTargetAddrTable (RFC-2573 - SNMP-TARGET-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
snmpTargetAddrTDomain	snmpUDPDomain = snmpDomains.1
snmpTargetAddrTAddress (IP Address and UDP Port of the Notification Receiver)	OCTET STRING (6) Octets 1-4: <IP Address> Octets 5-6: <Port>
snmpTargetAddrTimeout	<Timeout> from the TLV
snmpTargetAddrRetryCount	<Retries> from the TLV
snmpTargetAddrTagList	If <Trap type> == 1, 2 or 4 "@config_trap" Else If <Trap type> = 3 or 5 "@config_inform"
snmpTargetAddrParams	"@config_n" (Same as snmpTargetAddrName value)
snmpTargetAddrStorageType	volatile
snmpTargetAddrRowStatus	active (1)

3.6.1.3 snmpTargetAddrExtTable

Create 1 row for each TLV element in the config file

Table 7. snmpTargetAddrExtTable

snmpTargetAddrExtTable (RFC-2576 - SNMP-COMMUNITY-MIB)	New Row
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
snmpTargetAddrTMask	<Zero length octet string>
snmpTargetAddrMMS	0

3.6.1.4 snmpTargetParamsTable

Create 1 row for each TLV element in the config file. If <Trap type> is 1, 2, or 3, or if the <Security Name> Field is zero-length, create the table as follows:

Table 8. snmpTargetParamsTable for <Trap type> 1, 2, or 3

snmpTargetParamsTable (RFC-2573 - SNMP-TARGET-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	If <Trap type> = 1 SNMPv1 (0) Else If <Trap type> = 2 or 3 SNMPv2c (1) Else if <Trap type> = 4 or 5 SNMPv3 (3)
snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	If <Trap type> = 1 SNMPv1 (1) Else If <Trap type> = 2 or 3 SNMPv2c (2) Else if <Trap type> = 4 or 5 USM (3) **NOTE: The mapping of SNMP protocol types to value here are different from snmpTargetParamsMPModel**
snmpTargetParamsSecurityName	"@config"
snmpTargetParamsSecurityLevel	noAuthNoPriv
snmpTargetParamsStorageType	volatile
snmpTargetParamsRowStatus	active (1)

If <Trap type> is 4 or 5, and the <Security Name. Field is non-zero length, create the table as follows:

Table 9. snmp TargetParamsTable for ,Trap type> 4 or 5

snmpTargetParamsTable (RFC-2573 - SNMP-TARGET-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	If <Trap type> = 1 SNMPv1 (0) Else If <Trap type> = 2 or 3 SNMPv2c (1) Else if <Trap type> = 4 or 5 SNMPv3 (3)
snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	If <Trap type> = 1 SNMPv1 (1) Else If <Trap type> = 2 or 3 SNMPv2c (2) Else if <Trap type> = 4 or 5 USM (3) **NOTE: The mapping of SNMP protocol types to value here are different from snmpTargetParamsMPModel**
snmpTargetParamsSecurityName	<Security Name>
snmpTargetParamsSecurityLevel	The security level of <Security Name>
snmpTargetParamsStorageType	volatile
snmpTargetParamsRowStatus	active (1)

3.6.1.5 snmpNotifyFilterProfileTable –

Create 1 row for each TLV that has a non-zero <Filter Length>

Table 10. snmpNotifyFilterProfileTable

snmpNotifyFilterProfileTable (RFC-2573 - SNMP-NOTIFICATION-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
snmpNotifyFilterProfileName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
snmpNotifyFilterProfileStorType	volatile
snmpNotifyFilterProfileRowStatus	active (1)

3.6.1.6 snmpNotifyFilterTable

Create 1 row for each TLV that has a non-zero <Filter Length>

Table 11. snmpNotifyFilterTable

snmpNotifyFilterTable (RFC-2573 - SNMP-NOTIFICATION-MIB	New Row
Column Name (* = Part of Index)	Column Value
* snmpNotifyFilterProfileName	"@config_n" Where n ranges from 0 to m-1 where m is the number of notification receiver TLV elements in the config file
* snmpNotifyFilterSubtree	<Filter OID> from the TLV
snmpNotifyFilterMask	<Zero Length Octet String>
snmpNotifyFilterType	included (1)
snmpNotifyFilterStorageType	volatile
snmpNotifyFilterRowStatus	active (1)

3.6.1.7 snmpCommunityTable

Create 1 row with fixed values if 1 or more TLVs is present

This causes SNMPV1 and V2c Notifications to contain the community string in snmpCommunityName

Table 12. snmpCommunityTable

snmpCommunityTable (RFC-2576 - SNMP-COMMUNITY-MIB	1st Row
Column Name (* = Part of Index)	Column Value
* snmpCommunityIndex	"@config"
snmpCommunityName	"public"
snmpCommunitySecurityName	"@config"
snmpCommunityContextEngineID	<The engineID of the cable modem>
snmpCommunityContextName	<Zero length octet string>
snmpCommunityTransportTag	<Zero length octet string>
snmpCommunityStorageType	volatile
snmpCommunityStatus	active (1)

3.6.1.8 usmUserTable

Create 1 row with fixed values, if 1 or more TLVs is present. Other rows are created, one each time the engine ID of a trap receiver is discovered

This specifies the user name on the remote notification receivers to send notifications to.

One row in the usmUserTable is created. Then when the engine ID of each notification receiver is discovered, the agent copies this row into a new row and replaces the 0x00 in the usmUserEngineID column with the newly discovered value.

Table 13. usmUserTable

usmUserTable (RFC-2574 - SNMP-USER-BASED-SM-MIB	1st Row
Column Name (* = Part of Index)	Column Value
* usmUserEngineID	0x00
* usmUserName	"@config" - When other rows are created, this is replaced with the <Security Name> field from the TLV element.
usmUserSecurityName	"@config" - When other rows are created, this is replaced with the <Security Name> field from the TLV element.
usmUserCloneFrom	<don't care> - can't clone this row
usmUserAuthProtocol	None - When other rows are created, this is replaced with None or MD5, depending on the security level of the V3 User
usmUserAuthKeyChange	<don't care> - write only
usmUserOwnAuthKeyChange	<don't care> - write only
usmUserPrivProtocol	None - When other rows are created, this is replaced with None or DES, depending on the security level of the V3 User
usmUserPrivKeyChange	<don't care> - write only
usmUserOwnPrivKeyChange	<don't care> - write only
usmUserPublic	<zero length string>
usmUserStorageType	volatile
usmUserStatus	Active (1)

3.6.1.9 vacmSecurityToGroupTable

Create 3 rows with fixed values, if 1 or more TLVs is present

These are the 3 rows with fixed values - These are used for the TLV entries with <Trap Type> set to 1, 2, or 3 or with a zero length <Security Name>

Table 14. vacmSecurityToGroupTable

vacmSecurityToGroupTable (RFC-2575 - SNMP-VIEW-BASED-ACM-MIB	1st Row	2nd Row	3rd Row
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmSecurityModel	SNMPV1 (1)	SNMPV2c (2)	USM (3)
* vacmSecurityName	"@config"	"@config"	"@config"
vacmGroupName	"@configV1"	"@configV2"	"@configUSM"
vacmSecurityToGroupStorageType	volatile	volatile	volatile
vacmSecurityToGroupStatus	active (1)	active (1)	active (1)

The TLV entries with <Trap Type> set to 4 or 5 and a non-zero length <Security Name> will use the rows created in the vacmSecurityToGroupTable by the DH Kickstart process.

3.6.1.10 vacmAccessTable

Create 3 rows with fixed values, if 1 or more TLVs is present

These are the 3 rows with fixed values - These are used for the TLV entries with <Trap Type> set to 1, 2, or 3 or with a zero length <Security Name>

Table 15. vacmAccessTable

vacmAccessTable (RFC-2575 - SNMP-VIEW-BASED-ACM-MIB	1st Row	2nd Row	3rd Row
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmGroupName	"@configV1"	"@configV2"	"@configUSM"
* vacmAccessContextPrefix	<Zero length string>	<Zero length string>	<Zero length string>
* vacmAccessSecurityModel	SNMPV1 (1)	SNMPV2c (2)	USM (3)
* vacmAccessSecurityLevel	noAuthNoPriv (1)	noAuthNoPriv (1)	noAuthNoPriv (1)
vacmAccessContextMatch	exact (1)	exact (1)	exact (1)
vacmAccessReadViewName	<Zero length octet string>	<Zero length octet string>	<Zero length octet string>
vacmAccessWriteViewName	<Zero length octet string>	<Zero length octet string>	<Zero length octet string>
vacmAccessNotifyViewName	"@config"	"@config"	"@config"
vacmAccessStorageType	volatile	volatile	volatile
vacmAccessStatus	active (1)	active (1)	active (1)

The TLV entries with <Trap Type> set to 4 or 5 and a non-zero length <Security Name> will use the rows created in the vacmAccessTable by the DH Kickstart process.

3.6.1.11 vacmViewTreeFamilyTable

Create 1 row with fixed values if 1 or more TLVs is present

This row is used for the TLV entries with <Trap Type> set to 1, 2, or 3 or with a zero length <Security Name>

Table 16. vacmViewTreeFamilyTable

vacmViewTreeFamilyTable (RFC-2575 - SNMP-VIEW-BASED-ACM-MIB	1st Row
Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	"@config"
* vacmViewTreeFamilySubtree	1.3
vacmViewTreeFamilyMask	<Default from MIB>
vacmViewTreeFamilyType	included (1)
vacmViewTreeFamilyStorageType	volatile
vacmViewTreeFamilyStatus	active (1)

The TLV entries with <Trap Type> set to 4 or 5 and a non-zero length <Security Name> will use the rows created in the vacmViewTreeFamilyTable by the DH Kickstart process.

4 OSSI for Radio Frequency Interface

4.1 Subscriber Account Management Interface Specification[39]

Note: The Subscriber Account Management Interface Specification is OPTIONAL for CMTS vendors at this time. However, if a billing interface is provided by a CMTS vendor, it MUST conform to the specification in this section.

The Subscriber Account Management Interface Specification is defined to enable prospective vendors of cable modems and cable modem termination systems to address the operational requirements of subscriber account management in a uniform and consistent manner. It is the intention that this would enable operators and other interested parties to define, design and develop Operations and Business Support System (OBSS) necessary for the commercial deployment of different class of services over cable networks with accompanying usage-based billing of services for each individual subscriber.

Subscriber Account Management described here refers to the following business processes and terms:

- Class of Service Provisioning Processes, which are involved in the automatic and dynamic provisioning and enforcement of subscribed class of policy-based service level agreements (SLAs);
- Usage-Based Billing Processes, which are involved in the processing of bills based on services rendered to and consumed by paying subscribers. This Specification focuses primarily on bandwidth-centric usage-based billing scenarios. It complements the current Telephony Billing Specification that is being developed within the PacketCable architecture.

In order to develop the DOCSIS-OSS Subscriber Account Management Specification, it is necessary to consider high-level business processes common to cable operators and the associated operational scenarios. These issues are discussed in Appendix B.

4.1.1 Service Flows, Service Classes, and Subscriber Usage Billing

The DOCSIS 1.1 RFI specification provides a mechanism for a Cable Modem (CM) to register with its Cable Modem Termination System (CMTS) and to configure itself based on external Quality of Service (QoS) parameters when it is powered up or reset. To quote (in part) from Section 8.1 Theory of Operation:

The principal mechanism for providing enhanced QoS is to classify packets traversing the RF MAC interface into a Service Flow. A Service Flow is a unidirectional flow of packets that is provided a particular Quality of Service. The CM and the CMTS provide this QoS by shaping, policing, and prioritizing traffic according to the QoS Parameter Set defined for the Service Flow.

The requirements for Quality of Service include:

- A configuration and registration function for pre-configuring CM-based QoS Service Flows and traffic parameters.
- Utilization of QoS traffic parameters for downstream Service Flows.
- Classification of packets arriving from the upper layer service interface to a specific active Service Flow
- Grouping of Service Flow properties into named Service Classes, so upper layer entities and external applications (at both the CM and the CMTS) can request Service Flows with desired QoS parameters in a globally consistent way.

A Service Class Name (SCN) is defined in the CMTS via provisioning (see *DOCS-QOS-MIB)*. An SCN provides a handle to an associated QoS Parameter Set (QPS) template. Service Flows that are created

[39] Section 4.1 replaced per ECN OSS-N-02197 by GO on 11/15/02.

using an SCN are considered to be "named" Service Flows. The SCN identifies the service characteristics of a Service Flow to external systems such as a billing system or customer service system. For consistency in billing, operators should ensure that SCNs are unique within an area serviced by the same BSS that utilizes this interface. A descriptive SCN might be something like *PrimaryUp*, *GoldUp, VoiceDn,* or *BronzeDn* to indicate the nature and direction of the Service Flow to the external system.

A Service Package implements a Service Level Agreement (SLA) between the MSO and its Subscribers on the RFI interface. A Service Package might be known by a name such as *Gold*, *Silver*, or *Bronze*. A Service Package is itself implemented by the set of named Service Flows (using SCNs) that are placed into a CM Configuration File[40] that is stored on a TFTP server. The set of Service Flows defined in the CM Config File are used to create active Service Flows when the CM registers with the CMTS. Note that many Subscribers are assigned to the same Service Package, therefore, many CMs use the same CM Config File to establish their active Service Flows. Also, note that a Service Package has to define at least two Service Flows known as Primary Service Flows that are used by default when a packet matches none of the classifiers for the other Service Flows. A CM Config File that implements a Service Package, therefore, must define the two primary Service Flows using SCNs (e.g. *PrimaryUp* and *PrimaryDn*) that are known to the CMTS if these Service Flows are to be visible to external systems via this billing interface. Note that it is often the practice in a usage sensitive billing environment to segregate the operator's own maintenance traffic to and from the CM into the primary service flows so that this traffic is not reflected in the traffic counters associated the subscriber's SLA service flows.

The DOCSIS 1.1 RFI specification also provides for dynamically created Service Flows. An example could be a set of dynamic Service Flows created by an embedded PacketCable Multimedia Terminal Adapter (MTA) to manage VoIP signaling and media flows. All dynamic Service Flows must be created using an SCN known to the CMTS if they are to be visible to the billing system. These dynamic SCNs do not need to appear in the CM Config File but the MTA may refer to them directly during its own initialization and operation.

During initialization, a CM communicates with a DHCP Server that provides the CM with its assigned IP address and, in addition, receives a pointer to the TFTP Server that stores the assigned CM Config File for that CM. The CM reads the CM Config File and forwards the set of Service Flow definitions (using SCNs) up to the CMTS. The CMTS then performs a macro-expansion on the SCNs (using its provisioned SCN templates) into QoS Parameter Sets sent in the Registration Response for the CM. Internally, each active Service Flow is identified by a 32-bit SFID assigned by the CMTS to a specific CM (relative to the RFI interface). For billing purposes, however, the SFID is not sufficient as the only identifier of a Service Flow because the billing system cannot distinguish the class of service being delivered by one SFID from another. Therefore, the SCN is necessary, in addition to the SFID, to identify the Service Flow's class of service characteristics to the billing system. The billing system can then rate the charges differently for each of the Service Flow traffic counts based on its Service Class (e.g. Gold octet counts are likely to be charged more than Bronze octet counts). Thus, the billing system obtains from the CMTS the traffic counts for each named Service Flow (identified by SFID and SCN) that a subscriber's CM uses during the billing data collection interval. This is true even if multiple active Service Flows (i.e. SFIDs) are created using the same SCN for a given CM over time. This will result in multiple billing records for the CM for Service Flows that have the same SCN (but different SFIDs). Note that the SFID is the primary key to the Service Flow. When an active Service Flow exists across multiple sequential billing files the SFID allows the sequence of recorded counter values to be correlated to the same Service Flow instance.

[40] The CM Configuration File contains several kinds of information needed to properly configure the CM and its relationship with the CMTS, but for the sake of this discussion, only the Service Flow and Quality of Service components are of interest.

4.1.2 IP Detail Record (IPDR) Standard

The IPDR Organization (see www.ipdr.org) has defined a generic model for using XML Schema in IP Detail Recording applications. Industry specific IP billing applications such as the Cable Data Systems Subscriber Usage Billing Record can be added to the IPDR standard by mapping the application semantics onto the NDM-U XML Schema syntax. See Appendix E for the DOCSIS OSSI Service Specification submission to IPDR.org for the *DOCSIS Cable Data Systems Subscriber Usage Billing Record*. Appendix E also contains an example IPDR XML format Subscriber Usage Billing file and the IPDR standard XML Schema (.xsd) files that describe the DOCSIS IPDR syntax.

4.1.2.1 IPDR Network Model

The IPDR Network Model is given in the *NDM-U 3.1* specification and is portrayed in Figure 2 below. Note that in Figure 2 the highlighted blocks and interfaces are the only ones defined in this specification. In this network model, the Service Consumer (SC**)** is the Cable Data Service Subscriber identified by their Cable Modem MAC address, current CM IP address, and current CPE IP addresses. The Service Element (SE) is the CMTS identified by its host name, IP address, and current value of its sysUpTime object. The IPDR Recorder (IR) is the billing record formatter function that creates the NDM-U 3.1 schema format XML IPDRs from the internal counters maintained by the CMTS for each Subscriber's running and terminated Service Flows. The IPDR Store (IS) is the function that maintains the billing file in the FTP file system and detects that the billing file has been deleted by the billing collector. The IPDR Recorder and the IPDR Store are functions that may be implemented within the CMTS or hosted on another platform such as an Element Management System (EMS) or Record Keeping Server (RKS). The IPDR Transmitter (IT) represents the billing record collectors that retrieve the billing records from the IPDR Store as specified in section 4.1.5. In this specification the IT retrieves the compressed and possibly encrypted billing file from the IS on a collection cycle determined by the IT.

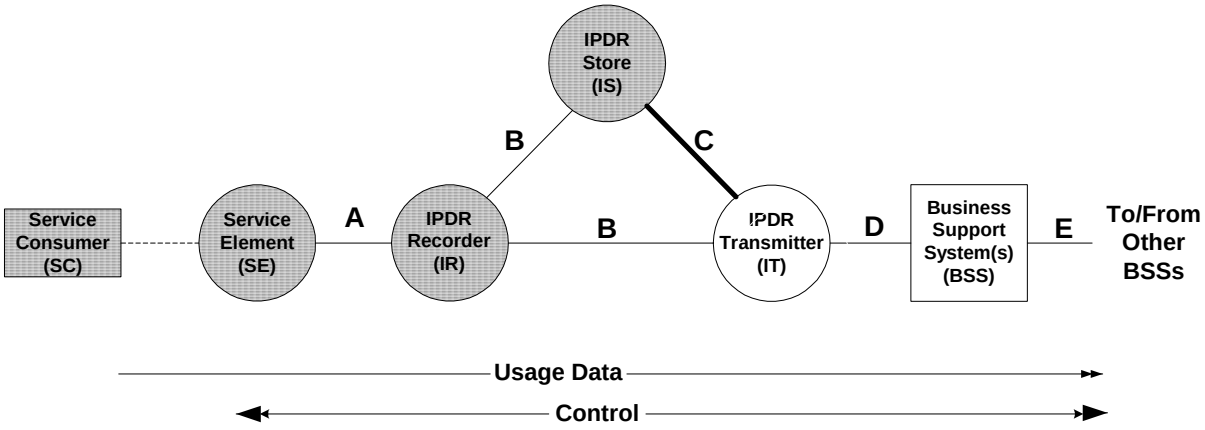

Figure 2. Basic Network Model (ref. NDM-U 3.1 from www.ipdr.org)

Note that the A-interface is not specified by the NDM-U specification because it is an internal interface between the SE and the IR components. The B-interface between the IR and the IS component is also internal to the implementation and is not specified here. In addition, the other B-interface between the IR and the IT components is not used by this specification and is outside the scope of this specification. The C-interface is specified by the NDM-U specification as a file of IPDR records formatted according to the IPDRdoc XML Schema (.xsd) files (see Appendix E). In addition, the billing file in the C-interface is compressed as required by section 4.1.5. The C-interface billing file MUST be implemented using the *DOCSIS Cable Data Systems Subscriber Usage Billing Record* submission to the IPDR standard as defined in Appendix E. The D- and E-interfaces are beyond the scope of this specification.

4.1.3 High-Level Requirements for Subscriber Usage Billing Records

This section provides the high-level, functional requirements of this interface. Use of spec words is intentionally avoided as subsequent sections will specify the actual requirements necessary for interoperability utilizing this interface.

The CMTS, or its supporting Element Management System (EMS), must provide formatted Subscriber Usage Billing Records for all subscribers attached to the CMTS on demand to a mediation system or a billing system. The minimum billing record collection interval that must be supported by a CMTS is 15 minutes. The following are the requirements for processing and transmitting Subscriber Usage Billing Records:

1. The Subscriber Usage Billing File must identify the CMTS by host name and IP address and the time that the billing file was created. The sysUpTime value for the CMTS must also be recorded.
2. Subscriber usage billing records must be identified by CM MAC address (but not necessarily sorted). The Subscriber's current CM IP address must also be present in the billing record for the Subscriber. If the CMTS is tracking CPE IP addresses behind the Subscriber's CM, then these CPE IP addresses must also be present in the billing record.
3. Subscriber usage billing records must have entries for each active Service Flow (identified by SFID and Service Class Name) used by all CMs operating in DOCSIS 1.1 (or higher) registration mode during the collection interval[41]. This includes all currently running Service Flows as well as all terminated Service Flows that were deleted and logged during the collection interval. Note well that a provisioned or admitted state SF that was deleted before it became active is not recorded in the billing file, even though it was logged by the CMTS. In addition, billing records for CMs operating in DOCSIS 1.0 registration mode may be created by reporting the DOCSIS 1.0 service as a pair of upstream and downstream Service Flows that contain the aggregate packet and octet counters for each direction. In this case, the billing record must identify the CM as operating in 1.0 mode. Note that there will be null Service Class Names associated with these DOCSIS 1.0 Service Flows.
4. It must be possible to distinguish running Service Flows from terminated Service Flows in the billing records. Internal CMTS Service Flow log records must not be deleted from the CMTS until after they have been recorded in a billing file stored in non-volatile storage. The CMTS must maintain a separate view of the internal Service Flow log for SNMP access via the DOCS-QOS-MIB. It must not be possible to delete internal Service Flow log entries via SNMP until they have been released by the billing formatter. A terminated Service Flow must be reported into a Billing File exactly once.
5. It must be possible to identify the Service Flow direction as upstream or downstream without reference to the Service Class Name. The number of packets and octets passed must be collected for each upstream and downstream Service Flow. The number of packets dropped and the number of packets delayed due to enforcement of QoS maximum throughput parameters (SLA) must also be collected for each Service Flow. In the case of an upstream Service Flow, the reported SLA drop and delay counters must represent only the policing performed by the CMTS. Note that since it is possible for a Subscriber to change from one service package to another and back again or to have dynamic service flows occur multiple times, it is possible that there will be multiple entries for a given SCN within a Subscriber's billing record for the collection period. This could also occur if a CM re-registers for any reason (such as CM power failure).
6. All traffic counters must be based on absolute 64-bit counters as maintained by the CMTS. These counters must be reset to zero by the CMTS if it reinitializes its management interface. The CMTS sysUpTime value is used to determine if the management interface has been reset between

[41] Subscriber billing records are a method of byte usage accounting only. Some types of Service Flows can consume system resources without bytes actually being passed (e.g. an active RTPS flow or an admitted UGS flow). Billing for these types of resources is beyond the scope of this specification.

adjacent collection intervals. It is expected that the 64-bit counters will not roll over within the service lifetime of the CMTS.

7. To facilitate processing of the Subscriber Usage Billing Records by a large number of diverse billing and mediation systems an Extensible Markup Language (XML) format is required. Specifically, the IP Detail Record (IPDR) standard as described in IPDR.org's *Network Data Management – Usage, Version 3.1* (NDM-U 3.1) as extended for XML schema format DOCSIS Cable Data Systems Subscriber Usage Billing Records must be used. See Appendix E for the *DOCSIS Cable Data Systems Subscriber Usage Billing Records* Service Specification submission to IPDR.org, the DOCSIS IPDR schema, and an example DOCSIS IPDR XML Schema billing file. See also http://www.ipdr.org for more information on the NDM-U specification and Service Specification Guidelines.
8. To improve the performance of storage and transmission of the NDM-U XML format billing records a compressed file format is required. Lossless compression in GZIP 4.3 format as described in RFC-1952 must be used to store and transmit the billing file. It is expected that an IPDRv3 XML format billing file will compress on the order of 30:1 or better. See also http://www.gnu.org/software/gzip for more information.
9. To improve the network performance of the billing collection activity, a reliable high-throughput TCP stream must be used to transfer billing records between the record formatter and the collection system. Standard FTP GET of the compressed (and optionally encrypted) billing file from the record formatter by the collection system must be supported.
10. To allow for decoupled scheduling, the billing collection cycle must be driven by the collection system through the standard FTP GET and FTP DELETE operations. Since the collection interval may vary over time, the record formatter is only required to maintain one current billing file in its FTP file system. The collection system (operating on its own schedule) may retrieve the current billing file using FTP GET at any time after it has been constructed and placed in the FTP file system by the record formatter. The collection system must explicitly FTP DELETE the billing file when it no longer needs it. The retrieval model is detailed in Section 4.1.5.
11. To ensure the end-to-end privacy and integrity of the billing records, while either stored or in transit, an authentication and encryption mechanism must be provided between the record formatter and the collection system. The security model is detailed in Section 4.1.6.

4.1.4 Billing Collection Interval

Subscriber Usage Billing Records report the absolute traffic counter values for each Service Flow used by a Cable Modem (Subscriber) that has become active during the billing collection interval as seen at the end of the interval. The collection interval is defined as the time between the creation of the previous billing file (Tprev) and the creation of the current billing file (Tnow). See Figure 3 below. There are two kinds of Service Flows that are reported in the current billing file: 1) SFs that are still running at the time the billing file is created and 2) terminated SFs that have been deleted and logged during the collection interval. A provisioned or admitted state SF that was deleted before it became active MUST NOT be recorded in the billing file, even though it was logged by the CMTS.

The CMTS (or supporting EMS) MUST record any currently running SFs using Tnow as the timestamp for its counters and MUST identify them in the IPDR SFtype element as "Interim". Terminated SFs that have a deletion time (Tdel) later than Tprev are the only ones recorded in the current billing file (i.e. a terminated SF MUST BE reported exactly once). A CMTS MUST record a terminated SF using its Tdel from the log as the timestamp for its counters and MUST identify it in the IPDR SFtype element as "Stop". Note that the timestamps are based on the formatter's recording times, not the collection system's retrieval times. Since the collection cycle may vary over time, the recording times in the billing file can be used to construct an accurate time base over sequences of billing files.

In the example shown in Figure 3 below there are four Service Flows recorded for a Subscriber in the current billing file being created at Tnow. SFa is a long running SF that was running during the previous collection interval (it has the same SFID in both the current and the previous billing files). SFa was recorded as type Interim at Tprev in the previous billing file and is recorded again as type Interim at Tnow in the current file. SFb is a running SF that was created during the current collection interval. SFb is recorded as type Interim for the first time at Tnow in the current file. SFc is a terminated SF that was running during the previous collection interval but was deleted and logged during the current collection interval. SFc was recorded as type Interim at Tprev in the previous billing file and is recorded as type Stop at the logged Tdel(c) in the current file. SFd is a terminated SF that was both created and deleted during the current collection interval. SFd is recorded only once as type Stop at the logged Tdel(d) in the current billing file only.

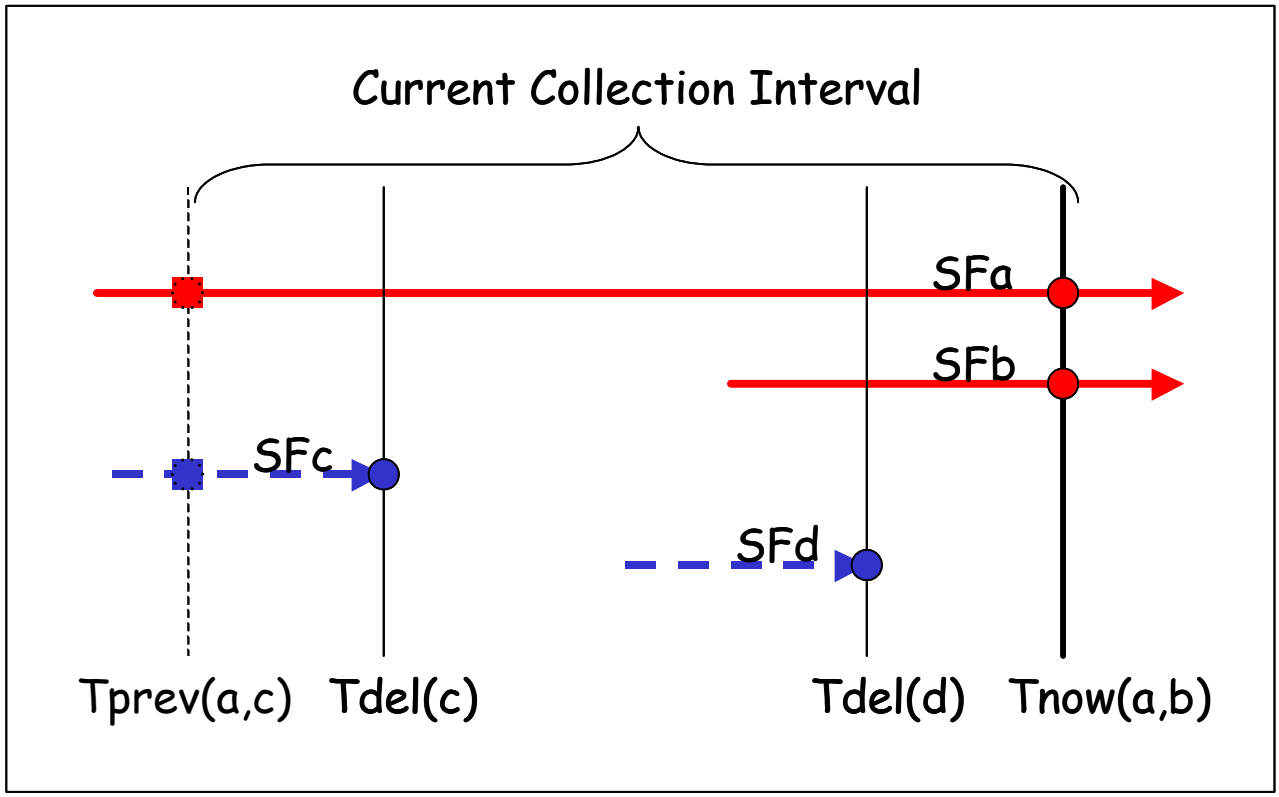

Figure 3. Billing Collection Interval Example

4.1.5 Billing File Retrieval Model

Billing files are built by the record formatter on the CMTS (or supporting EMS) and are then retrieved by the collection system in a decoupled manner using FTP semantics. There is no explicit signaling protocol between them and no prior arrangement regarding the frequency of billing collection. The CMTS (or supporting EMS) is responsible for creating the current billing file and MUST place it into its FTP file system only when the file is completely built. The formatter only creates one billing file which it MUST protect until the collection system is done with it. The collection system MAY retrieve the current billing file via FTP GET at any time after the file becomes available in the formatter's FTP file system. When the collection system has successfully retrieved the billing file, it MUST remove the file via FTP DELETE from the formatter's FTP file system. The formatter MUST monitor the existence of the billing file in its FTP file system and when it no longer exists, the formatter MUST begin to create the next billing file. The formatter

MUST finish constructing the next billing file and have it ready for retrieval in its FTP file system within 15 minutes of the previous file's deletion. If the billing file does not yet exist in the formatter's FTP file system when the collection system comes to retrieve it, the collection system MUST back off and return later to try again. The specific timeout for collection system retries is implementation dependent, however, the collection system MUST NOT make more than 3 retrieval attempts within any 5-minute period.

Note that if the collection system fails for any reason, the formatter will retain and protect the last billing file created until the collection system returns to retrieve the file. In this case, even though the recording timestamps in the current billing file may be quite old, the collection system will still retrieve the current file and delete it in the standard manner. The formatter will then immediately begin construction of a new billing file based on the current values of the CMTS's internal absolute 64-bit counters and the current timestamp. The collection system may then return at any time after the minimum cycle time (i.e. 15 minutes) and retrieve the new billing file with the current timestamps. The absolute values of the counters will always be preserved by the CMTS while it is operating, only the collection interval will be extended due to the outage on the collection system. The billing system can use the recording timestamps in the two files to accurately reconstruct the time base of the counters. Furthermore, the collection system MAY deliberately vary its collection cycles based on time of day or day of week. This decoupled billing file retrieval model works well for this case also.

The decoupled billing file retrieval model also supports multiple retrievals by multiple collection systems so long as the last collection system deletes the billing file when it is done with it. However, there is no requirement to support multiple simultaneous file transfers from the formatter. How the multiple collection systems coordinate this between themselves is beyond the scope of this specification

4.1.6 Billing File Security Model

The billing file security model has two components: 1) secure user authentication to control access to the billing file in the formatter's FTP file system and 2) secure file transfer to ensure the privacy and the integrity of the billing file while it is in transit. Both of these components are provided by the Secure Shell protocol version 2 (SSH2) and its Secure FTP (SFTP) subsystem as described by Internet drafts maintained by the IETF's SECSH working group at www.ietf.org/html.charters/secsh-charter.html. Additional information may be obtained from www.openssh.org, which provides an open source implementation of SSH2 and SFTP. A CMTS (or supporting EMS) hosting the billing formatter MUST provide secure access to its FTP file system via SSH2 and SFTP. It is also strongly recommended that the operator disable legacy insecure Telnet and FTP access to the formatter's platform when SSH2/SFTP are active. How legacy Telnet and FTP are disabled is beyond the scope of this specification.

To ensure restricted access to billing information, the billing collector MUST have it's own userid and password for access to the formatter's billing file directory via SSH2/SFTP. Furthermore, the billing collector's userid MUST NOT be shared with any other applications or users hosted on the formatter's platform. SSH2 user public key authentication is OPTIONAL for the billing collector's userid. How userids, keys, and passwords are administered on the formatter's platform is beyond the scope of this specification. Note also that the collection system requires both read and delete access permissions to the billing file directory in the formatter's FTP file system.

While the formatter's platform MUST provide secure authentication and file transfer capabilities, the operator may elect to not utilize them. In this case, the formatter's platform MUST provide access to the billing file directory via legacy insecure FTP and the billing collector MUST have its own userid and password for legacy FTP access as well. Again, it is strongly recommended that the operator not allow insecure legacy FTP access to the formatter's billing file

4.1.7 IPDR Record Structure

The NDM-U 3.1 specification defines the IPDRDoc record structure. The IPDRDoc 3.1 XML schema (see IPDRDoc3.1.xsd in Appendix E) defines the hierarchy of elements within the IPDR document that MUST be supported by the CMTS (as shown in Figure 4 below).

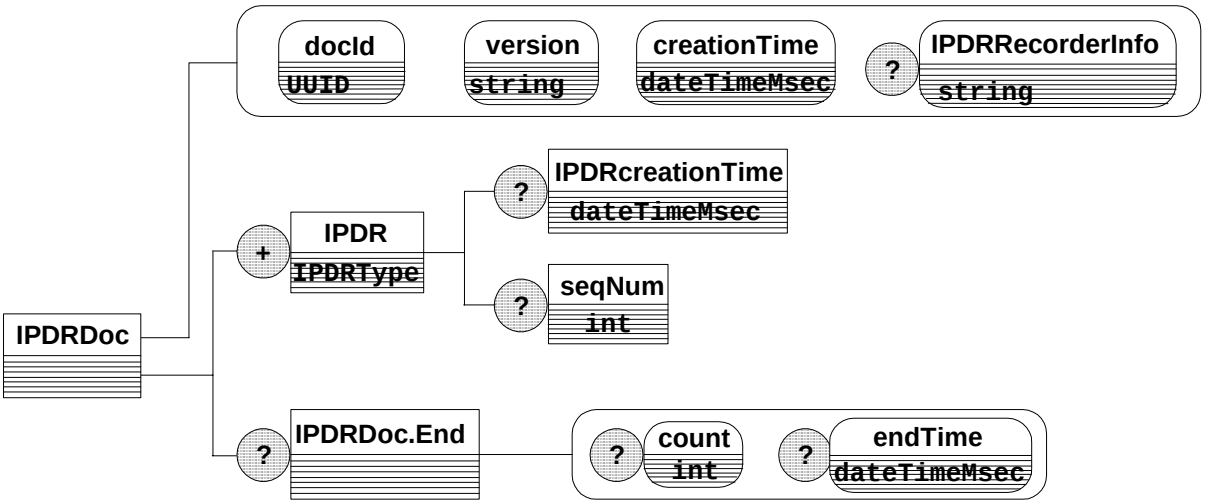

Figure 4. IPDRDoc 3.1 Generic Schema

The IPDRDoc3.1.xsd schema defines the generic structure of any IPDR document regardless of application. To complete the definition of an application specific IPDR record structure, an application schema must be provided that imports the basic IPDRDoc3.1.xsd schema. The DOCSIS IPDR Version 3.1 schema (see DOCSIS-3.1-B.0.xsd in Appendix E.2) defines the elements that record the DOCSIS specific information that MUST be supported by the CMTS (as shown in Figure 5 below). Note that the DOCSIS-Type in is the application specific implementation of the IPDR element shown in Figure 4 above. Thus, the DOCSIS specific elements are sub elements of the IPDR element.

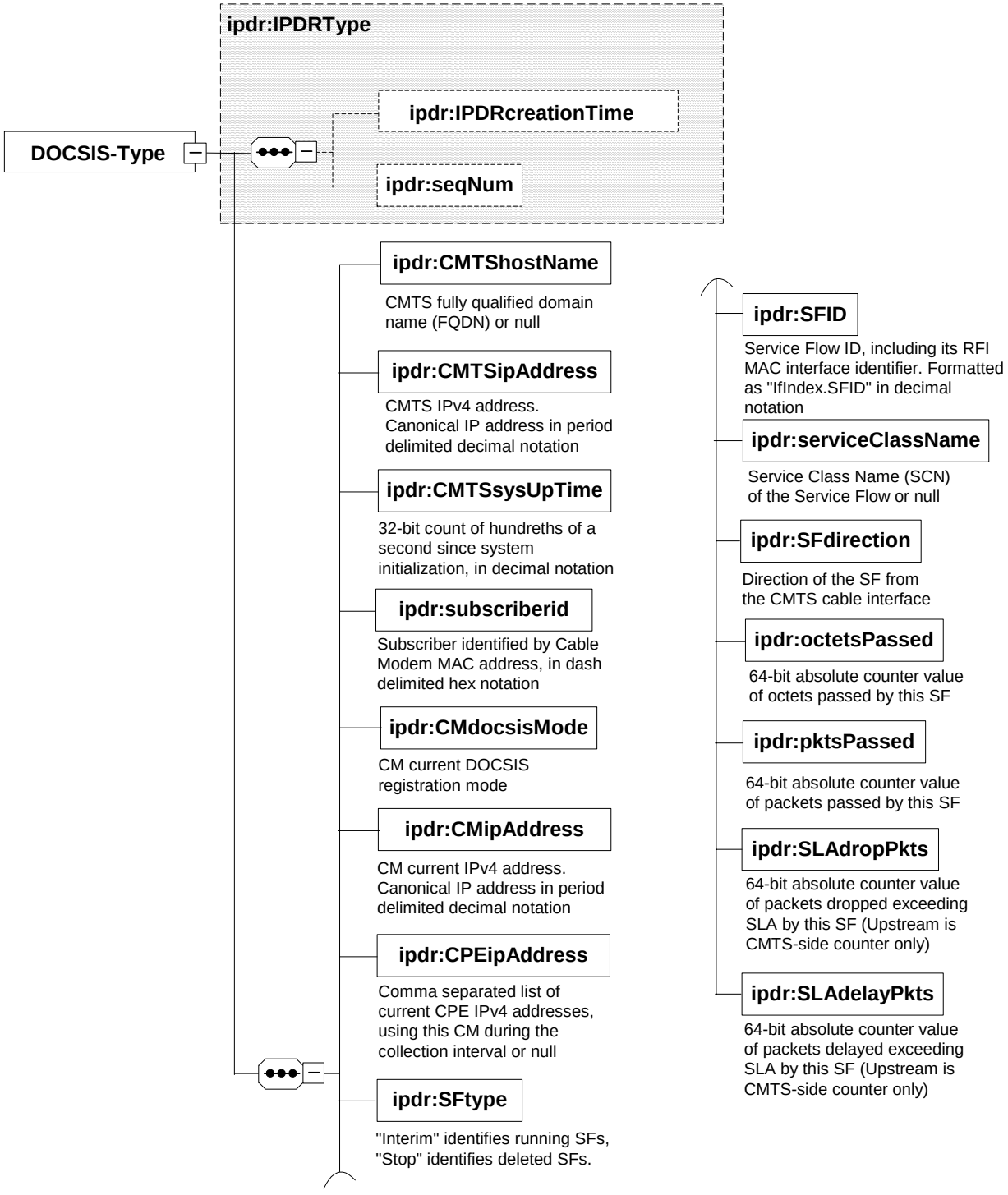

Figure 5. DOCSIS IPDR 3.1 Schema

The following elements and attributes are the only ones used by the DOCSIS Cable Data Systems Subscriber Usage Billing Record IPDR instance document (see Appendix E). These elements and attributes are described below:

1. The *IPDRDoc* element is the outermost element that describes the IPDR billing file itself. It defines the XML namespace, the identity of the XML schema document, the version of the specification,

the timestamp for the file, a unique document identifier, and the identity of the IPDR recorder. An IPDRDoc is composed of multiple IPDR records. The attributes for the IPDRDoc element MUST be as follows:

a) *xmlns*=”http://www.ipdr.org/namespaces/ipdr”
Constant: the XML namespace identifier. Defined by ipdr.org.

b) *xmlns:xsi*="http://www.w3.org/2001/XMLSchema-instance"
Constant: the XML base schema identifier. Defined by ipdr.org.

c) *xsi:schemaLocation*="DOCSIS-3.1-B.0.xsd"
Constant: the name of the DOCSIS application specific schema file.

d) *version*=“3.1”
Constant: the version of the IPDR document. Defined by ipdr.org.

e) *creationTime* =”yyyy-mm-ddThh:mm:ssZ”
UTC time stamp at the time the billing file is created (in ISO format). For example: creationTime="2002-06-12T21:11:21Z". Note that IPDR timestamps MUST always be in UTC/GMT (Z).

f) *docId*="<32-bit UTC timestamp>-0000-0000-0000-<48-bit MAC address>"
The unique document identifier. The DOCSIS docId is in a simplified format that is compatible with the Universal Unique Identifier (UUID) format required by the IPDR NDM-U 3.1 specification. The 32-bit UTC timestamp component MUST be the IPDRDoc creationTime in seconds since the epoch 1 Jan 1970 UTC formatted as eight hex digits. The 48-bit MAC address component MUST be the ethernet address of the CMTS management interface formatted as 12 hex digits. All other components MUST be set to zero. In the context of the minimum 15-minute IPDR billing file collection cycle specified in this document, this simplified UUID is guaranteed to be unique across all CMTSs and for the foreseeable future. For example:
docId="3d07b8f9-0000-0000-0000-00015c11bfbe".

g) *IPDRRecorderInfo*=”hostname.mso.com”
Identifies the IPDR Recorder (IR) from the network model in Figure 2 above. This attribute MUST identify the billing record formatter by the fully qualified hostname of the CMTS or the EMS where the formatter resides. If a hostname is not available, then this MUST be the IPv4 address of the CMTS or EMS formatted in dotted decimal notation.

2. An *IPDR* element MUST describe a single Subscriber Usage Billing Record for a single DOCSIS service flow. The IPDR is further structured into DOCSIS specific sub elements that describe the details of the CMTS, the subscriber (CM and CPE), and the service flow itself. While the generic IPDR record structure is designed to describe most time-based and event-oriented IP services, this feature is not particularly relevant to the Cable Data Service Subscriber Usage Billing Records and is largely ignored. This is because a service session at the CMTS is just the aggregate usage of an active Service Flow during the billing collection interval. Another way to look at it is as if there is really only one event being recorded: the billing collection event itself. The attributes for the IPDR element are

    - *xsi:type*="DOCSIS-Type"
      Constant: identifies the DOCSIS application specific type of the IPDR record.

3. The *IPDRcreationTime* element identifies the time associated with the counters for this service flow. The format MUST be the same as the IPDRDoc creationTime attribute (see 1e. above). IPDRcreationTime MUST be the same as the IPDRDoc creationTime when the service flow is still running (i.e. SFtype = Interim). IPDRcreationTime MUST be the time the service flow was deleted when the service flow has been terminated (i.e. SFtype = Stop). Note that a Stop IPDR is always earlier than the IPDRDoc creationTime. Also, note that this sub element is optional in the basic IPDR 3.1 schema, but is REQUIRED for all DOCSIS IPDRs.

4. The *seqNum* element is an optional sub element of the basic IPDR 3.1 schema. It MUST NOT be used in DOCSIS IPDRs. Note that there is no ordering implied in DOCSIS IPDRs within an IPDRDoc.
5. The *CMTShostName* element is a REQUIRED element that contains the fully qualified domain name (FQDN) of the CMTS if it exists. For example: cmts01.mso.com. This element MUST be null if no FQDN exists (i.e. <CMTShostName></CMTShostName> or <CMTShostName/>).
6. The *CMTSipAddress* element contains the IP address of the management interface of the CMTS. This element is REQUIRED and MUST be represented in standard IPv4 decimal dotted notation (for example: 10.10.10.1).
7. The *CMTSsysUpTime* element contains the value of the sysUpTime SNMP object in the CMTS taken at the IPDRDoc creationTime. This element is REQUIRED and MUST be the count of 100ths of seconds since the CMTS management interface was initialized. If the CMTSsysUpTime regresses between adjacent IPDRDocs, then the CMTS management interface has been reset and all service flow counters have been reset to zero. Note well: this value MUST be the same for each IPDR within a given IPDRDoc file, regardless of the IPDRcreationTime of a given IPDR.
8. The *subscriberId* element contains the unique identifier of the subscriber. This element is REQUIRED and MUST be the subscriber's cable modem 48-bit MAC address formatted as dash delimited hex digits. For example: 11-11-11-11-11-11.
9. The CMdocsisMode element identifies the registration mode of the Cable Modem as "1.0", "1.1", or "2.0". If the registration mode is "1.0" then the reported Service Flow contains the aggregate packet and octet counters for the DOCSIS 1.0 service in this direction. This element is REQUIRED.
10. The *CMipAddress* element contains the current IP address of the subscriber's cable modem. This element is REQUIRED and MUST be represented in standard IPv4 decimal dotted notation (for example, 10.100.100.123). Note that this address can change over a set of IPDRDoc files if the operator's DHCP server reassigns IP addresses to cable modems.
11. The *CPEipAddress* element MUST contain a comma delimited list of the current IP addresses of all of the subscriber's CPE using this cable modem or null if there are none being tracked by the CMTS (i.e. <CPEipAddress></CPEipAddress> or <CPEipAddress/>) . If there are multiple CPE using the CM, then there MUST be multiple CPE IP addresses in the list. Each CPE IP address MUST be represented in standard IPv4 decimal dotted notation (for example: 12.12.12.123 or 12.12.12.123, 12.12.12.124, 12.12.12.125). Note that the configuration state of the DOCS-SUBMGT-MIB influences whether CPE IP addresses are being tracked by the CMTS and are thus being reported in the IPDRs (the DOCS-SUBMGT-MIB controls the CM and CPE filters on the CMTS).
12. The *SFtype* element identifies the kind of service flow being described by this IPDR. This element is REQUIRED and MUST have either of two values: "Interim" identifies this SF as currently running in the CMTS and "Stop" identifies this SF as having been terminated in the CMTS. A running service flow has active counters in the CMTS and this IPDR MUST contain the current sample of these counters. A terminated service flow has logged counters in the CMTS and this IPDR MUST contain the final counter values for this service flow. Note well: the internal logged SF counters on the CMTS MUST NOT be deleted until after the terminated service flow has been recorded into an IPDR record that has been stored in non-volatile memory, regardless of any other capability to manage them via SNMP through the DOCS-QOS-MIB.
13. The *SFID* element contains the internal service flow identifier known to the CMTS. This element is REQUIRED and is needed to correlate the IPDRs for an individual service flow between adjacent IPDRDoc files when computing delta counters between samples. Note that SFIDs are relative to their RFI MAC interface. Therefore, the SFID element MUST be formatted as ifIndex.SFID where the ifIndex component is the interface index in the CMTS ifTable for the RFI MAC interface and the SFID component is the32-bit identifier assigned by the CMTS to this service flow. Both components MUST be represented as decimal values (for example, 15.34567). To avoid potential confusion in the billing system, the CMTS MUST NOT reuse the SFID component for a minimum of two billing collection cycles.

14. The *serviceClassName* element contains the name associated with the QoS parameter set for this service flow in the CMTS. The SCN is an ASCII string identifier, such as “GoldUp” or “SilverDn”, that can be used by external operations systems to assign, monitor, and bill for different levels of bandwidth service without having to interpret the details of the QoS parameter set itself. A service flow is associated with an SCN whenever a cable modem configuration file uses the SCN to define an active service flow. A dynamic service flow application such as PacketCable may also assign an SCN to a service flow as a parameter during the dynamic creation of the service flow. Note that use of SCNs is optional within the context of the DOCSIS RFI specification, however, for operational purposes, especially when billing for tiered data services per this specification, their use often becomes mandatory. Since this policy is within the control of the operator, the use of SCNs is not mandatory in this specification, but rather highly recommended. Note well: this element is REQUIRED in the IPDR record, but if no SCN is used to identify the service flow in the CMTS, then this element MUST have a null value (that is <serviceClassName></serviceClassName> or <serviceClassName/>). Note also that a CM operating in DOCSIS 1.0 mode will not have any SCNs assigned and this element will be null.
15. The *SFdirection* element identifies the service flow direction relative to the CMTS RFI interface. This element is REQUIRED and MUST have one of two values: “Upstream” identifies service flows passing packets from the cable modem to the CMTS, and “Downstream” identifies service flows passing packets from the CMTS to the cable modem.
16. The *octetsPassed* element MUST contain the current 64-bit count of the number of octets passed by this service flow formatted in decimal notation. This element is REQUIRED. If the SFtype is Interim, then this is the current value of the running counter. If the SFtype is Stop, then this is the final value of the terminated counter. The 64-bit counter value will not wrap around within the service lifetime of the CMTS. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate octet count for the DOCSIS 1.0 service in this direction.
17. The *pktsPassed* element MUST contain the current 64-bit count of the number of packets passed by this service flow formatted in decimal notation. This element is REQUIRED. If the SFtype is Interim, then this is the current value of the running counter. If the SFtype is Stop, then this is the final value of the terminated counter. The 64-bit counter value will not wrap around within the service lifetime of the CMTS. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate packet count for the DOCSIS 1.0 service in this direction.
18. The *SLAdropPkts* and *SLAdelayedPkts* elements contain the current 64-bit count of the number of packets dropped or delayed by this service flow due to enforcement of the maximum throughput limit specified by the Service Level Agreement (SLA) as implemented by the QoS parameter set. These elements are REQUIRED for all service flows. For upstream service flows, these counters record only the SLA enforcement performed by the CMTS. Upstream packets dropped or delayed at the CM are not recorded here. These counters are formatted in decimal notation. If the SFtype is Interim, then this is the current value of the running counter. If the SFtype is Stop, then this is the final value of the terminated counter. The 64-bit counter value will not wrap around within the service lifetime of the CMTS. If the CMdocsisMode for this service flow is “1.0” then these elements contain the aggregate SLA policing packet count for the DOCSIS 1.0 service in this direction. Note that these values are provided to aid the operator in identifying subscribers who are attempting to use more bandwidth than their SLA provides. This may be an opportunity to offer the subscriber a higher capacity SLA consistent with his/her demonstrated needs.
19. *IPDRDoc.End* MUST be the last element inside IPDRDoc that describes the IPDR billing file itself. It defines the count of IPDRs that are contained in the file and the ending timestamp for the file creation.
    a) *count*=”nnnn”
       Where nnnn MUST be the decimal count of the number of IPDR records in this IPDRDoc.
    b) *endTime* =”yyyy-mm-ddThh:mm:ssZ”
       MUST be the UTC time stamp at the time the billing file is completed (formatted as above). For example:endTime=" 2002-06-12T21:11:23Z".

4.2 Configuration Management

Configuration management is concerned with initializing, maintaining, adding and updating network components. In a DOCSIS environment, this includes a cable modem and/or CMTS. Unlike performance, fault, and account management, which emphasize network monitoring, configuration management is primarily concerned with network control. Network control, as defined by this interface specification, is concerned with modifying parameters in and causing actions to be taken by the cable modem and/or CMTS. Configuration parameters could include both identifiable physical resources (for example, Ethernet Interface) and logical objects (for example, IP Filter Table).

Modifying the configuration information of a CM and/or CMTS can be categorized as follows:

- Non-operational
- Operational

Non-operational changes occur when a manager issues a modify command to a CM/CMTS, and the change doesn't effect the operating environment. For example, a manager may change contact information, such as the name and address of the person responsible for a CMTS.

Operational changes occur when a manager issues a modify command to a CM/CMTS, and the change affects the underlying resource or environment. For example, a manager may change the docsDevResetNow object from false to true, which in turn will cause the CM to reboot.

To adjust the necessary attribute values, the CM and CMTS MUST support MIB objects as specified in section 3 of this document.

While the network is in operation, configuration management will be responsible for monitoring the configuration and making changes in response to commands via SNMP or in response to other network management functions.

For example, a *performance management function* may detect that response time is degrading due to a high number of uncorrected frames, and may issue a configuration management change to modify the modulation type from 16Qam to QPSK. A *fault management function* may detect and isolate a fault and may issue a configuration management change to bypass the fault.

4.2.1 Version Control

The CM MUST support software revision and operational parameter configuration interrogation.

The CM MUST include at least the hardware version, Boot ROM image version, vendor name, software version, and model number in the sysDescr object (from [RFC-3418]). The CM MUST support docsDevSwCurrentVers MIB object and the object MUST contain the same software revision information as shown in the software information included in the sysDescr object.

The format of the specific information contained in the sysDescr MUST be as follows:

To report	**Format of each field**
Hardware Version	HW_REV: <Hardware version>
Vendor Name	VENDOR: <Vendor name>
Boot ROM	BOOTR: <Boot ROM Version>
Software Version	SW_REV: <Software version>
Model Number	MODEL: <Model number>

Each type value pair MUST be separated with a colon and blank space. Each pair is separated by a ";" followed by a blank. For instance, a sysDescr of a CM of vendor X, hardware version 5.2, Boot ROM version 1.4, SW version 2.2, and model number X

MUST appear as following:

any text<<HW_REV: 5.2; VENDOR: X; BOOTR: 1.4; SW_REV 2.2; MODEL: X>>any text

The CM MUST report at least all of the information necessary in determining what SW the CM is capable of being upgraded to. If any fields are not applicable, the CM MUST report "NONE" as the value. For example; CM with no BOOTR, CM will report BOOTR: NONE.

The CM MUST implement the docsDevSwCurrentVers object ([RFC-2669]) to report the current software version.

The intent of specifying the format of sysDescr is to define how to report information in a consistent manner so that sysDescr field information can be programmatically parsed. This format specification does not intend to restrict the vendor's hardware version numbering policy.

The CMTS MUST implement the sysDescr object (from [RFC-3418]). For CMTS, format of information and the content of the information in sysDescr is vendor dependent.

4.2.2 System Initialization and Configuration

There are several methods available to configure CM and CMTS including console port, SNMP set, configuration file, and configuration-file-based SNMP encoded object. The CM MUST support system initialization and configuration via configuration file, configuration-file-based SNMP encoded object and SNMP set. The CMTS MUST support system initialization and configuration via telnet connection, console port, and SNMP set. The CM and CMTS (only CMTS that support configuration by configuration file) MUST support any valid configuration file regardless of configuration file size.

4.2.3 Secure Software Upgrades

The CM secure software upgrade detail process is documented in the Appendix D of BPI+ specification.

DOCSIS 1.1 CM MUST use secure software upgrade mechanism to perform software upgrade regardless of what DOCSIS CMTS version (1.0 or 1.1) it is connected to. When a 1.1 CM is connected to a 1.1 CMTS, the 1.1 CM MUST operate in either DOCSIS 1.1 mode or DOCSIS 1.0 mode. When a 1.1 CM is connected to a 1.0 CMTS, the 1.1 CM MUST operate in DOCSIS 1.0 mode. This means that a DOCSIS 1.1 CM MUST use secure software upgrade mechanism to perform software upgrade regardless of what mode it operates in (1.0 mode or 1.1 mode).

There are two available secure software download schemes including manufacture control scheme and operator control scheme.

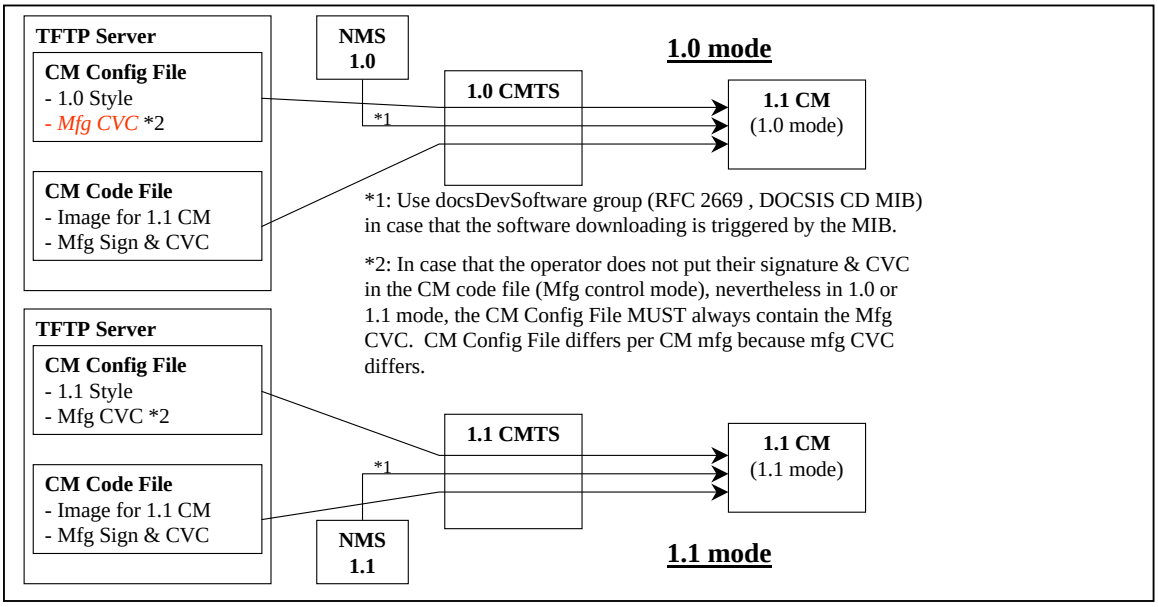

Figure 6. Manufacture control scheme

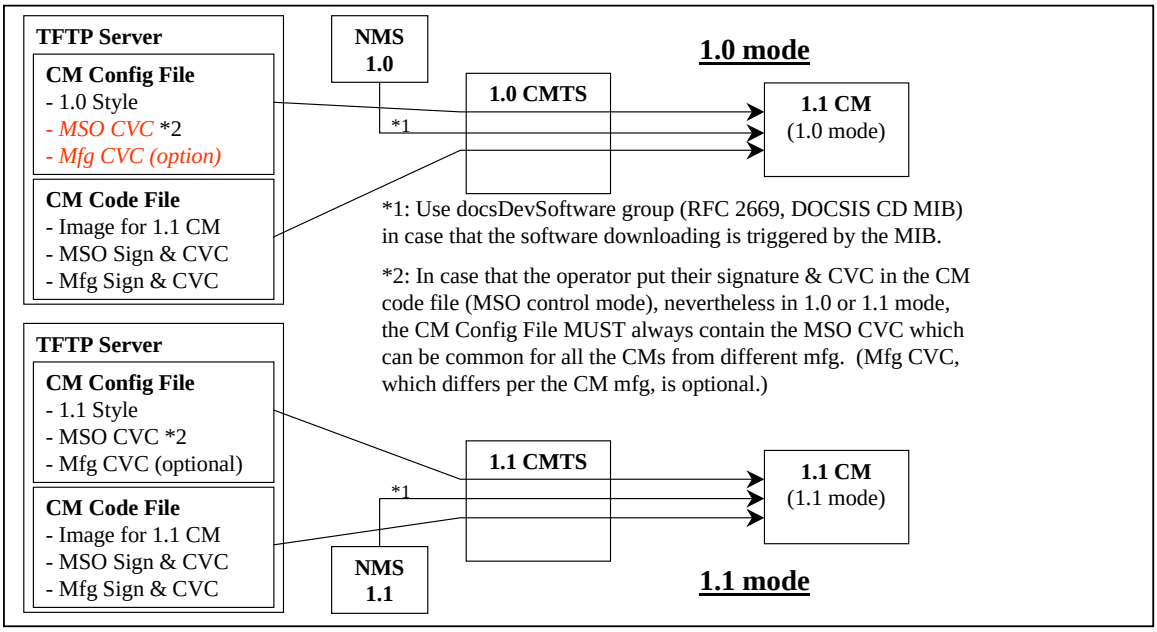

Figure 7. Operator control scheme.

Prior to secure software upgrade initialization, CVC information is needed to be initialized at the CM for software upgrade. Depending on the scheme (described above) that the operator chooses to implement, appropriate CVC information MUST be include in the configuration file. It is recommended that CVC information always be present in the configuration file so that a device will always have the CVC information initialized and read if the operator decides to use SNMP-initiate upgrade as a method to trigger a secure software upgrade operation. If the operator decides to use configuration-file-initiate upgrade as a method to trigger secure software download, CVC information is needed to be present in the configuration file at the time the modem is rebooted to get the configuration file that will trigger the upgrade only.

There are two methods to trigger secure software download including SNMP-initiated and configuration-file-initiated. Both methods MUST be supported by CM and MAY be supported by CMTS.

The following describes the SNMP-initiated mechanism. Prior to SNMP-initiate upgrade, a CM MUST have valid X.509 compliant code verification certificate information. From a network management station:

- Set docsDevSwServer to the address of the TFTP server for software upgrades
- Set docsDevSwFilename to the file pathname of the software upgrade image
- Set docsDevSwAdminStatus to Upgrade-from-mgt.

If docsDevSwAdminStatus is set to ignoreProvisioningUpgrade(3), the CM MUST ignore any software download configuration file setting and not attempt a configuration file initiated upgrade.

docsDevSwAdminStatus MUST persist across reset/reboots until over-written from an SNMP manager or via a TLV-11 setting in the CM configuration file.

The default state of docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2} until it is over-written by ignoreProvisioningUpgrade{3} following a successful SNMP initiated software upgrade or otherwise altered by the management station.

docsDevSwOperStatus MUST persist across resets to report the outcome of the last software upgrade attempt.

After the CM has completed a configuration-file-initiated secure software upgrade, the CM MUST reboot and become operational with the correct software image as specified in [DOCSIS 5]. After the CM is registered, it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MAY be the filename of the software currently operating on the CM
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the CM
- docsDevSwOperStatus MUST be completeFromProvisioning{2}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the CM

After the CM has completed an SNMP-initiated secure software upgrade, the CM MUST reboot and become operational with the correct software image as specified in [DOCSIS 5]. After the CM is registered, it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be ignoreProvisioningUpgrade{3}
- docsDevSwFilename MAY be the filename of the software currently operating on the CM
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the CM
- docsDevOperStatus MUST be completeFromMgt{3}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the CM

The CM MUST properly use ignoreProvisioningUpgrade status to ignore software upgrade value that may be included in the CM configuration file and become operation with the correct software image and after the CM is registered, it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be ignoreProvisioningUpgrade{3}
- docsDevSwFilename MAY be the filename of the software currently operating on the CM
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the CM
- docsDevSwOperStatus MUST be completeFromMgt{3}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the CM

Retries due to a power loss or reset are only required for an SNMP-initiated upgrade. If a power loss or reset occurs during a config file-initiated upgrade, the CM will follow the upgrade TLV directives in the configuration file upon reboot. It will not retry the previous upgrade. The config file upgrade TLVs essentially provides a retry mechanism that is not available for an SNMP-initiated upgrade.

If a CM suffers a loss of power or resets during an SNMP-initiated upgrade, the CM MUST resume the upgrade without requiring manual intervention and when the CM resumes the upgrade process:

- docsDevSwAdminStatus MUST be Upgrade-from-mgt{1}
- docsDevSwFilename MUST be the filename of the software image to be upgraded
- docsDevSwServer MUST be the address of the TFTP server containing the software upgrade image to be upgraded
- docsDevSwOperStatus MUST be inProgress{1}
- docsDevSwCurrentVers MUST be the current version of software that is operating on the CM

In case where the CM reaches the maximum number of TFTP download retries (max retries = 3) resulting from multiple losses of power or resets during an SNMP-initiated upgrade, the CM MUST behave as specified in [DOCSIS 5]; in addition, the CM's status MUST adhere to the following requirements after it is registered:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade process.
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the CM

If a CM suffers a loss of power or resets during a configuration file-initiated upgrade, when the CM reboots the CM MUST ignore the fact that a previous upgrade was in progress and either not perform an upgrade if no upgrade TLVs are present in the config file, or if upgrade TLVs are present take the action described in the requirements in section 10.1 of [DOCSIS 5], at the time of the reboot.

In the case where the CM had a configuration file initiated upgrade in progress during a reset and if there are no upgrade TLVs in the config file upon reboot:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MAY be the filename of the current software image.
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating in the CM.
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVers MUST be the current version of software that is operating on the CM

In the case where the CM had a configuration file initiated upgrade in progress during a reset, if there are upgrade TLVs in the config file upon reboot:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename contained in TLV-9 of the config file.
- docsDevSwServer MUST be the address of the TFTP server containing the software to be loaded into the CM, (either the value of TLV-21 in the config file if present, or the address of the configuration file TFTP server if TLV-21 is not present per the requirements stated in section 10.1 of [DOCSIS 5].)
- docsDevSwOperStatus MUST be inProgress{1}
- docsDevSwCurrentVers MUST be the current version of software that is operating on the CM

If a CM exhausts the required number of TFTP retries by issuing a total of 16 consecutive TFTP requests, the CM MUST behave as specified in [DOCSIS 5] and then the CM MUST fall back to last known working image and proceed to an operational state and adhere to the following requirements:

- docDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docDevSwFilename MUST be the filename of the software that failed the upgrade process
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be failed{4}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the CM

In the case where CM successfully downloads (or detects during download) an image that is not intended for the CM device, the CM MUST behave as specified in [DOCSIS 5], section 10.1 “Downloading Cable Modem Operating Software” and adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the CM

In the case where CM determines that the download image is damaged or corrupted, the CM MUST reject the newly downloaded image. The CM MAY re-attempt to download if the maximum number of TFTP download retries (max retries = 3) has not been reached. If the CM chooses not to retry, the CM MUST fall back to the last known working image and proceed to an operational state, generate appropriate event notification as specified in Appendix F, and adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the CM

In the case where CM determines that the image is damaged or corrupted, the CM MUST reject the newly downloaded image. The CM MAY re-attempt to download the new image if the maximum number of TFTP download retries (max retries = 3) has not been reached. On the third consecutive failed retry of the CM software download attempt, the CM MUST fall back to the last known working image and proceed to an operational state. In this case, the CM MUST send two notifications, one to notify that the max retry limit has been reached, and another to notify that the image is damaged. Immediately after the CM reaches the operational state the CM MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the CM

4.3 Protocol Filters

The CM MUST implement LLC, SNMP Access, and IP protocol filters. The LLC protocol filter entries can be used to limit CM forwarding to a restricted set of network-layer protocols (such as IP, IPX, NetBIOS, and AppleTalk). The IP protocol filter entries can be used to restrict upstream or downstream traffic based on source and destination IP addresses, transport-layer protocols (such as TCP, UDP, and ICMP), and source and destination TCP/UDP port numbers.

CM MUST apply filters (or more properly, classifiers) in an order appropriate to the following layering model; specifically, the inbound MAC (or LLC) layer filters are applied first, then the "special" filters, then the IP layer inbound filters, then the IP layer outbound filters, then any final LLC outbound filters. Note that LLC outbound filters are expected future requirements of the DOCS-CABLE-DEVICE-MIB.[42]

4.3.1 LLC Filter

Inbound LLC filters, from docsDevFilterLLCTable, MUST be applied to layer-2 frames entering the CM from either the CATV MAC interface{2} and/or any CM CPE interface.

The object docsDevFilterLLCUnmatchedAction MUST apply to all (CM) interfaces. The default value of the (CM) docsDevFilterLLCUnmatchedAction MUST be set to accept.

docsDevFilterLLCUnmatchedAction:

If (CM docsDevFilterLLCUnmatchedAction is) set to discard(1), any L2 packet that does not match any LLC filters will be discarded, otherwise accepted. If (CM docsDevFilterLLCUnmatchedAction is) set to accept, any L2 packet that does not match any LLC filters will be accepted, otherwise discarded.

Another way to interpret this is the following:

```
action = UnMatchedAction
Iterate through the table
    if there is a match (packet.protocol = row.protocol)
       {
         reverse the action (accept becomes discard, discard becomes accept)
         apply action to the packet
         terminate the iteration
       }
```

LLC (CM) filters MUST apply to in-bound traffic direction only. Traffic generated from CM MUST not be applied to LLC filters (i.e. ARP requests, SNMP responses).

The CM MUST support a minimum of ten LLC protocol filter entries.

4.3.2 Special Filter

Special filters are IP spoofing filters and SNMP access filters. IP spoofing filters MUST only be applied to packets entering the CM from CMCI interface(s). SNMP access filters are in effect when the CM is not running in SNMPv3 agent mode and can be applied to both CMCI and CATV interfaces.

According to the interface number section of document, CMCI interface is a generic reference to any current or future form of CM CPE interface port technology.

[42] Revised requirement per ECN OSS-N-03066 by GO on 07/10/03.

4.3.3 IP Spoofing Filter

DOCSIS 1.1 CM MAY implement IP spoofing filter specified in RFC-2669.

If a CM supports the IP Spoofing filter functionality specified in RFC-2669, the CM MUST adhere to the following requirements:

- Implement all MIB objects in the docsDevCpeGroup
- Default value of docsDevCpeIpMax = -1

4.3.3.1 Additional requirement on dot1dTpFdbTable (RFC-1493)

CM CPE MAC addresses learned via CM configuration file MUST set the dot1dTpFdbStatus to “mgmt”. It is assumed that the number of “mgmt” configured CM CPE MAC addresses is <= to the TLV-18 (Maximum Number of CPE) value.

4.3.4 SNMP Access Filter

The SNMP access filters MUST be applied to SNMP packets entering from any interfaces and destined for the CM. SNMP access filter MUST be applied after IP spoofing filters for the packets entering the CM from the CMCI interface. Since SNMP access filter function is controlled by docsDevNmAccessTable, SNMP access filter is available and applies only when the CM is in SNMP v1/v2c NmAccess mode.

When CM is running in SNMP Coexistence mode SNMP access MUST be controlled and specified by MIB Objects in [RFC-3411-3415 and RFC-2576].[43]

4.3.4.1 docsDevNMAccessIP and docsDevNMAccessIpMask

The device that implement docsDevNmAccessTable applies the following rules in order to determine whether to permit SNMP access from a SrcIpAddr:

1. If (docsDevNmAccessIp == “255.255.255.255”), the CMTS/CM MUST permit the access from any SrcIpAddr.
2. If ((docsDevNmAccessIp AND docsDevNmAccessIpMask) == (SrcIpAddr AND docsDevNmAccessIpMask)), the CMTS/CM MUST permit the access from SrcIpAddr.
3. If neither #1 and #2 is applied, the CMTS/CM MUST NOT permit the access from SrcIpAddr.

The CMTS/CM’s default value of the docsDevNmAccessIpMask MUST be set to “0.0.0.0”.

The following are examples of the MIB values and the access.

docsDevNmAccessIp	docsDevNmAccessIpMask	Access
“255.255.255.255”	Any IP Address Mask	Any NMS
Any IP Address	“0.0.0.0”	Any NMS
Any IP Address except “255.255.255.255”	“255.255.255.255”	Single NMS
“0.0.0.0”	“255.255.255.255”	No NMS

[43] Revised RFC reference per ECN OSS-N-03066 by GO on 07/10/03.

4.3.5 IP Filter

The object docsDevFilterIPDefault MUST apply to all (CM) interfaces. DOCSIS 1.1 compliant CM MUST support a minimum 16 IP filters.

4.4 Fault Management

The goals of fault management are remote monitoring/detection, diagnosis, and correction of problems. Network Management operators rely on the ability to monitor and detect problems(s) (such as ability to trace and identify faults, accept and act on error-detection events), as well as the ability to diagnose and correct problem(s) (such as perform a sequences of diagnostic tests, correct faults, and display/maintain event logs.)

This section defines what MUST be available to support remote monitoring/detection, diagnosis and correction of problems.

4.4.1 SNMP Usage

In the DOCSIS environment, the goals of fault management are the remote detection, diagnosis, and correction of network problems. Therefore, the standalone CM MUST support SNMP management traffic across both the CPE and CATV MAC interfaces regardless of the CM's connectivity state. CCCMs MAY ignore the CPE management traffic, and MUST support SNMP on the CATV MAC interface once connectivity to CMTS is established. CM SNMP access may be restricted to support policy goals. CM installation personnel can use SNMP queries from a station on the CMCI side to perform on-site CM and diagnostics and fault classification (note that this may require temporary provisioning of the CM from a local DHCP server). Further, future CMCI side customer applications, using SNMP queries, can diagnose simple post-installation problems, avoiding visits from service personnel and minimizing help desk telephone queries.

Standard mib-2 [44]support MUST be implemented to instrument interface status, packet corruption, protocol errors, etc. The transmission MIB for Ethernet-like objects [RFC-2665] MUST be implemented on each cable device (CMTS/CM) Ethernet and Fast Ethernet port. Each cable device (CMTS/CM) MUST implement the ifXTable [RFC-2863] to provide discrimination between broadcast and multicast traffic.

The cable device (CMTS) MUST implement the extended version of MIB object docsIfCmtsCmStatusValue of ([DOCS-RFI-MIB]) as follows:[45]

```
docsIfCmtsCmStatusValue OBJECT-TYPE
    SYNTAX      INTEGER {
            other(1),
            ranging(2),
            rangingAborted(3),
            rangingComplete(4),
            ipComplete(5),
            registrationComplete(6),
            accessDenied(7),
            operational(8), --deprecated
```

[44] Revised text per ECN OSS-N-03066 by GO on 07/10/03.

[45] Added this paragraph and following statements per ECN OSS-N-03068 by GO on 07/11/03.

```
            registeredBPIInitializing(9)
        }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Current Cable Modem connectivity state, as specified
         in the RF Interface Specification. Returned status
         information is the CM status as assumed by the CMTS,
         and indicates the following events:
         other(1)
            Any state other than below.
         ranging(2)
            The CMTS has received an Initial Ranging Request
            message from the CM, and the ranging process is not  yet
            complete.
          rangingAborted(3)
            The CMTS has sent a Ranging Abort message to the CM.
          rangingComplete(4)
            The CMTS has sent a Ranging Complete message to the CM.
          ipComplete(5)
            The CMTS has received a DHCP reply message and forwarded
            it to the CM.
          registrationComplete(6)
            The CMTS has sent a Registration Response message to the CM.
          accessDenied(7)
            The CMTS has sent a Registration Aborted message
            to the CM.
          operational(8)   -- deprecated value
            If Baseline Privacy is enabled for the CM, the CMTS
            has completed Baseline Privacy initialization. If Baseline
            Privacy is not enabled, equivalent to registrationComplete.
          registeredBPIInitializing(9)
            Baseline Privacy is enabled, CMTS is in the process of
            completing the Baseline Privacy initialization. This state
            can last for a significant time in the case of failures
            during The process. After Baseline Privacy initialization
            Complete, the CMTS will report back the value
            registrationComplete(6).

            The CMTS only needs to report states it is able to detect."
        REFERENCE
          "Data-Over-Cable Service Interface Specifications: Radio
           Frequency Interface Specification SP-RFIv2.0-IO2-020617,
           Section 11.2."
```

```
        ::= { docsIfCmtsCmStatusEntry 9 }
```

The cable device (CMTS) MAY implement the new MIB object docsIfCmtsCmStatusValueLastUpdate in ([DOCS-IF-MIB]) as follows:

```
  docsIfCmtsCmStatusValueLastUpdate OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime when docsIfCmtsCmStatusValue was last updated"
::= { docsIfCmtsCmStatusEntry 22 }
```

The cable device (CMTS/CM) MUST support managed objects for fault management of the PHY and MAC layers. The DOCS-IF-MIB includes variables to track PHY state such as codeword collisions and corruption, signal-to-noise ratios, transmit and receive power levels, propagation delays, micro-reflections, in channel response, and Sync loss. The DOCS-IF-MIB also includes variables to track MAC state, such as collisions and excessive retries for requests, immediate data transmits, and initial ranging requests.

For fault management at all layers, the cable device (CMTS/CM) MUST generate replies to SNMP queries (subject to policy filters) for counters and status. The cable device (CMTS/CM) MUST send SNMP traps to one or more trap NMSs (subject to policy), and MUST send SYSLOG events to a SYSLOG server (if a SYSLOG server is defined).

When the cable device (CM) is operating in SNMP v1/v2c NmAccess mode it MUST support the capability of sending traps as specify by the following MIB object (proposed MIB extension to the docsDevNmAccess table):

```
DocsDevNmAccessTrapVersion OBJECT-TYPE
    SYNTAX         INTEGER {
      DisableSNMPv2trap(1),
      EnableSNMPv2trap(2),
     }
     MAX-ACCESS  read-create
     STATUS      current
     DESCRIPTION
         "Specifies the TRAP version that is sent to this NMS. Setting this
         object to DisableSNMPv2trap (1) causes the trap in SNMPv1 format to
         be sent to particular NMS. Setting this object to EnableSNMPv2trap
         (2) causes the trap in SNMPv2 format be sent to particular NMS"
     DEFVAL { Disable SNMPv2trap }
     ::= { docsDevNmAccessEntry 8 }
```

Any cable device (CMTS/CM) SHOULD implement the ifTestTable [RFC-2863] for any diagnostic test procedures that can be remotely initiated.

4.4.2 Event Notification

A cable device (CMTS/CM) MUST generate asynchronous events that indicate malfunction situations and notify about important non-fault events. Events could be stored in CMTS/CM device internal event LOG file, in non-volatile memory, get reported to other SNMP entities (as TRAP or INFORM SNMP messages), or be sent as a SYSLOG event message to a pre-defined SYSLOG server. Events MAY also be sent to the cable device (CMTS/CM) console; as a duplicate (identical) message to the optional console destination.

Event notification implemented by a cable device (CMTS/CM) MUST be fully configurable, by priority class; including the ability to disable SNMP Trap, SYSLOG transmission, and local logging. CMTS/CM MUST implement docsDevEvControlTable to control reporting of event classes. The object docsDevEvReporting MUST be implemented as RW for CMTS/CM.

A cable device (CMTS/CM) MUST support the following event notification mechanisms (regardless of what SNMP mode the cable device is in):

- local event logging
- SNMP TRAP/INFORM (trap-versions/targets/limiting/throttling)
- SYSLOG (targets/limiting/throttling)

Refer to the following sections for event notification implementation details.

When a CM is in SNMP v1/v2c NmAccess mode, the CM MUST support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (trap-versions/targets/limiting/throttling) as specified in RFC-2669 and OSSI 1.1. When CM is in SNMP coexistence mode, CM MUST support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669 and OSSI 1.1, and SNMP notification functions as specified in RFC-3413.[46]

If the CMTS supports, and is in SNMP v1/v2c NmAccess mode, the CMTS MUST support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669 and OSSI 1.1; however, SNMP TRAP (trap-versions/targets) MAY be implemented as specified in RFC-2669 and OSSI 1.1, or vendor proprietary MIB. When CMTS is in SNMP Coexistence mode, CMTS MUST support event notification functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669 and OSSI 1.1, and SNMP notification functions as specified in RFC-3413. [47]

4.4.2.1 Local Event Logging

A CM MUST maintain local-log events in both local-volatile storage and local-nonvolatile storage. A CMTS MUST maintain local-log events in local-volatile storage or local-nonvolatile storage or both. CMTS/CM events designated for local-volatile storage MAY also be retained in local-nonvolatile storage. CMTS/CM events designated for local-nonvolatile storage MAY also be retained in local-volatile storage. Data from local-volatile log and local-nonvolatile log is reported through docsDevEventTable. A DOCSIS 1.1 compliant cable device (CM/CMTS) MUST support the docsDevEvControlTable with additional requirements as described in this specification.

The cable device (CM/CMTS) local-log event-table MUST be organized as a cyclic buffer with a minimum of ten entries. CM/CMTS local-log data designated for local-nonvolatile storage MUST persist across reboots. The local-log event-table MUST be accessible through the cable device (CM/CMTS) docsDevEventTable [RFC-2669].

Aside from the procedures defined in this document, event recording must conform to the requirements of RFC-2669. Event descriptions must appear in English and must not be longer than 255 characters, which is the maximum defined for SnmpAdminString.

Events are identical if their EventIds are identical. For identical events occurring consecutively, the CM MAY choose to store only a single event. In such a case, the event description recorded MUST reflect the most recent event.

[46] Revised RFC reference per ECN OSS-N-03066 by GO on 07/10/03.

[47] Revised RFC reference per ECN OSS-N-03066 by GO on 07/10/03.

The EventId digit is a 32 bit unsigned integer. EventIds ranging from 0 to ((2^31) – 1) are reserved by DOCSIS. The EventId MUST be converted from the error codes defined in Appendix H (as updated by OSS-N-00108).

The EventIds ranging from 2^31 to ((2^32)-1) MUST be used as vendor specific EventIds using the following format:

- Bit 31 set to indicate vendor specific event
- Bits 30-16 contain bottom 15 bits of vendor's SNMP enterprise number
- Bits 15-0 used by vendor to number their events

Section 4.4.2.2.2 describes rules to generate unique EventIds from the error code.

RFC-2669 object docsDevEvIndex provides relative ordering of events in the log. The creation of local-volatile and local-nonvolatile logs necessitates a method for synchronizing docsDevEvIndex values between the two local logs after reboot. The following procedure MUST be used after reboot:

- The values of docsDevEvIndex maintained in the local non-volatile log MUST be renumbered beginning with 1.
- The local volatile log MUST then be initialized with the contents of the local non-volatile log.
- The first event recorded in the new active session's local- volatile log MUST use as its docsDevEvIndex the value of (last restored non-volatile docsDevEvIndex + 1).

A reset of the log initiated through an Snmp SET of RFC-2669 object docsDevEvControl MUST clear both the local-volatile and local-nonvolatile logs.

4.4.2.2 Format of Events

The Appendix H of this document lists all DOCSIS events.

The following sections explain the details how to report these events in any of the three mechanisms: local event logging, SNMP trap and syslog.

4.4.2.2.1 SNMP TRAP/INFORM[48]

A cable device (CMTS/CM) MUST send the following generic SNMP traps, as defined in standard MIB [RFC-1907] and [RFC-2863]:

- coldStart (warmStart is optional) [RFC-3418]
- linkUp [RFC-2863]
- linkDown [RFC-2863]
- SNMP authentication-Failure [RFC-3418]

A cable device (CMTS/CM) MUST implement SNMP traps defined in the DOCS-CABLE-DEVICE-TRAP-MIB, which is complementary to existing standard DOCSIS MIB-s (DOCS-CABLE-DEVICE-MIB, DOCS-BPI2-MIB,[49] and DOCS-IF-MIB) and defined in Appendix L.

- CM/CMTS in SNMP V1/V2c NmAccess mode MUST support SNMPv1 and SNMPv2c Traps.
- CM/CMTS in SNMP Coexistence mode MUST support SNMPv1, SNMPv2c, and SNMPv3 Traps.
- Cable device (CMTS/CM) MUST support INFORM.

[48] Revised various references per ECN OSS-N-03066 by GO on 07/10/03.

[49] Revised standard per ECN OSS-N-03066 by GO on 07/10/03.

INFORM is a variation of trap and requires the receiving host to acknowledge the arrival of an InformRequest-PDU with an InformResponse-PDU. An InformRequest-PDU is exactly the same as a trap-PDU except that the value in the PDU-type field is 6 for InformRequest-PDU instead of 7 for SNMPv2-trap-PDU. SNMPv1 does not support INFORM.

When a SNMP trap defined in the DOCS-CABLE-DEVICE-TRAP-MIB is enabled in a CM, it MUST send notifications for any event in its category whose priority is either “error” or “notice”. See the Table 17 in Section 4.4.2.3 "Standard DOCSIS Events for CM". It MAY notify (optionally) events with other priorities when it is possible.

When the SNMP trap defined in the DOCS-CABLE-DEVICE-TRAP-MIB is enabled in a CMTS, it MUST send notifications for an event whose priority is “critical” or “error” or “warning” or “notice”. See the Table 18, Table 19, and Table 20, in Section 4.4.2.4 "Standard DOCSIS Events for CMTS". It MAY send (optionally) events with other priorities.

Vendor-specific events reportable via SNMP TRAP MUST be described in the vendor documents. Vendor can also define vendor-specific SNMP traps and MUST do so in the private MIBs.

When defining vendor specific SNMP trap, the OBJECTS statement of the private trap definition SHOULD contain at least the objects explained below. For the CM traps, docsDevEvLevel, docsDevEvId, docsDevText, docsIfDocsisCapability, docsIfDocsisCapability, ifPhysAddress, and docsIfCmCmtsAddress SHOULD be included. For the CMTS traps, docsDevEvLevel, docsDevEvId, docsDevEvText, docsIfCmtsCmStatusDocsisMode, docsIfCmtsCmStatusMacAddress, docsIfDocsisOperMode, and ifPhysAddress SHOULD be included. For a description of the usage of these objects, please seek DOCS-CABLE-DEVICE-TRAP-MIB as reference. More objects may be contained in the OBJECTS body as desired.

Since the objects contained in these SNMP traps are the same objects in the SNMP local event table, CM MUST turn on the local event logging on a particular priority whenever the SNMP traps are configured on that event priority.

4.4.2.2.2 SYSLOG Message Format

For DOCSIS events, CM’s Syslog message MUST be sent in the following format and for non-DOCSIS events, it is optional.

*<level>*CABLEMODEM*[vendor]: <eventId> text vendor-specific-text*

For DOCSIS events, CMTS’s Syslog message MUST be sent in the following format and for non-DOCSIS events, it is optional.

<level>TIMESTAMP HOSTNAME CMTS*[vendor]: <eventId> text vendor-specific-text.*

Where:

- *Level* - ASCII presentation of the event priority, enclosed in angle brackets, which is constructed as OR of the default Facility (128) and event priority (0-7). The resulted level has the range between 128 and 135
- *TIMESTAMP and HOSTNAME* - MAY be sent after *<level>* by CMTS. If the TIMESTAMP and HOSTNAME fields are sent, they MUST be in the same format as the IETF proposed “draft-ietf-syslog-syslog-06.txt” TIMESTAMP and HOSTNAME format and MUST be sent together. The one space after TIMESTAMP is part of TIMESTAMP field. The one space after the HOSTNAME is part of HOSTNAME field
- *vendor* - Vendor name for the vendor-specific SYSLOG messages or DOCSIS for the standard DOCSIS messages.

- *EventId* - ASCII presentation of the INTEGER number in decimal format, enclosed in angle brackets, which uniquely identifies the type of event. This number MUST be the same number that is stored in docsDevEvId object in docsDevEventTable and also is associated with SNMP TRAP in the "SNMP TRAP/Inform" section.
- *text* – Vendor specific text.

For the standard DOCSIS events this number is converted from the error code using the following rules:

- The number is an eight digit decimal number.
- The first two digits (left most) are the ASCII code for the letter in the Error code.
- Next four digits are filled by 2 or 3 digits between the letter and the dot in the Error code with zero filling in the zap in the left side.
- The last two digits are filled by the number after the dot in the Error code with zero filling in the zap in the left.

For example, event D04.2 is converted into 68000402, and Event I114.1 is converted into 73011401.

Please note that this notion only uses a small portion of available number space reserved for DOCSIS (0 to 2^31-1). The first letter of an error code is always in upper case.

- *text* - for the standard DOCSIS messages this string MUST have the textual description as defined in [SP-OSSIv1.1 Appendix H]."
- *vendor-specific-text* - MAY be provided by vendors for vendor specific information.

There are products in the marketplace that expect existing syslog messages in their current format for fault management, which the DOCSIS syslog message format would break. So, for CM and CMTS, it is optional for the syslog message format of the non-DOCSIS events to follow the above formats.

The example of the syslog event for the event D04.2

"Time of the day received in invalid format":

<132>CABLEMODEM[DOCSIS]: <44000402> Time of Day Response but invalid data/format.

The number 44000402 in the given example is the number assigned by DOCSIS to this particular event.

4.4.2.3 Standard DOCSIS Events for CM[50]

The DOCS-CABLE-DEVICE-MIB defines 8 different priority levels and the corresponding reporting mechanism for each level. The standard DOCSIS events specified in this document utilizes the subset of these priority levels.

Emergency event (priority 1)

Reserved for vendor-specific 'fatal' hardware or software errors that prevents normal system operation and causes reporting system to reboot.

Every vendor may define there own set of emergency events. The examples of such events could be 'no memory buffers available', 'memory test failure' etc. (Such basic cross-vendor type events should be included in the DOCSIS 1.1 "Events for Notification" Appendix H so that vendors do not define many overlapping EventId's in vendor-private scope)

[50] Revised various standards in this section per ECN OSS-N-03066 by GO on 07/10/03.

Alert event (priority 2)

A serious failure, which causes reporting system to reboot but it is not caused by h/w or s/w malfunctioning. After recovering from the critical event system MUST send the cold/warm start notification. Alert event could not be reported as a Trap or SYSLOG message and MUST be stored in the internal log file. The code of this event MUST be saved in non-volatile memory and reported later through docsIfCmStatusCode SNMP variable DOCS-IF-MIB.

Critical event (priority 3)

A serious failure that requires attention and prevents the device from transmitting data but could be recovered without rebooting the system. After recovering from the error event Cable Modem Device MUST send the Link Up notification. Critical events could not be reported as a Trap or SYSLOG message and MUST be stored in the internal log file. The code of this event MUST be reported later through docsIfCmStatusCode SNMP variable DOCS-IF-MIB. The examples of such events could be configuration file problems detected by the modem or inability to get IP address from DHCP.

Error event (priority 4)

A failure occurred that could interrupt the normal data flow but does not cause modem to re-register. Error events could be reported in real time by using TRAP or SYSLOG mechanism.

Warning event (priority 5)

A failure occurred that could interrupt the normal data flow but does not cause modem to re-register. 'Warning' level is assigned to events both modem and CMTS have information about. So to prevent sending same event both from the CM and CMTS, trap and Syslog reporting mechanism is disabled by default for this level.

Notice event (priority 6)

The event of importance which is not a failure and could be reported in real time by using TRAP or SYSLOG mechanism. The examples of the NOTICE events are 'Cold Start', 'Warm Start', 'Link Up' and 'SW upgrade successful'. For a CM, an example of a Notice event is 'SW UPGRADE SUCCESS'[51]

Informational event (priority 7)

The not-important event, which is not failure, but could be helpful for tracing the normal modem operation. Local-Log messaging is allowed for vendor-specific informational events and subject to the constraints outlined in Section 2.2 of this document.[52]

Debug event (priority 8)

Reserved for vendor-specific non-critical events

The priority associated with the event is hard-coded and can't be changed. The reporting mechanism for each priority could be changed from the default reporting mechanism (Table 17) by using docsDevEvReporting object in DOCS-CABLE-DEVICE-MIB.

[51] Added last sentence per ECN OSS-N-02192, chg #5, by GO, on 12/06/02.

[52] Revised this event statement per ECN OSS-N-03045 by GO on 05/01/03.

Table 17. Default event priorities for the Cable Modem Device[53, 54]

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency	Yes	No	No	No or Yes*
2 Alert	Yes	No	No	No or Yes*
3 Critical	Yes	No	No	No or Yes*
4 Error	No or Yes**	Yes	Yes	Yes
5 Warning	No or Yes**	No	No	Yes
6 Notice	No or Yes**	Yes	Yes	Yes
7 Informational	No or Yes**	No	No	No
8 Debug	No	No	No	No

*Note: CMTS/CM events designated for local-nonvolatile storage MAY also be retained in local-volatile storage

**Note: CMTS/CM events designated for local-volatile storage MAY also be retained in local-nonvolatile storage.

Notifications for standard DOCSIS events generated by the CM MUST be in the format specified in Appendix H.

4.4.2.4 Standard DOCSIS Events for CMTS

CMTS uses the same levels of the event priorities as a CM; however, the severity definition of the events is different. Events with the severity level of Warning and less specify problems that could affect individual user (for example, individual CM registration problem).

Severity level of 'Error' indicates problems with a group of CMs (for example CMs that share same upstream channel).

Severity level of 'Critical' indicates problem that affects whole cable system operation, but is not a faulty condition of CMTS device. In all these cases CMTS MUST be able to send SYSLOG event and (or) SNMP TRAP to the NMS.

Severity level of 'Emergency' is vendor-specific and indicates problems with the CMTS hardware or software, which prevents CMTS operation.

[53] Deleted 'NOTE' column in Tables 17 through 22, per ECN OSS-N-02192 by GO, on 12/06/02.

[54] Revised table and added two footnotes per ECN OSS-N-03006 and ECN OSS-N-03045 by GO on 04/21/03 and 05/01/03.

Table 18. Default Event priorities for CMTS supporting only local-log non-volatile

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency	Yes	No	No	Not Used
2 Alert	Yes	No	No	Not Used
3 Critical	Yes	Yes	Yes	Not Used
4 Error	Yes	Yes	Yes	Not Used
5 Warning	Yes	Yes	Yes	Not Used
6 Notice	Yes	Yes	Yes	Not Used
7 Informational	No	No	No	Not Used
8 Debug	No	No	No	Not Used

A CMTS supporting only one local-log storage mechanism SHOULD accept any SNMP-Set operation on the optional docsDevEvReporting bit-value and always report value zero for the optional bit on SNMP-Get operations.

Table 19. Default Event priorities for CMTS supporting only local-log volatile

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency	Not Used	No	No	Yes
2 Alert	Not Used	No	No	Yes
3 Critical	Not Used	Yes	Yes	Yes
4 Error	Not Used	Yes	Yes	Yes
5 Warning	Not Used	Yes	Yes	Yes
6 Notice	Not Used	Yes	Yes	Yes
7 Informational	Not Used	No	No	No
8 Debug	Not Used	No	No	No

A CMTS supporting only one local-log storage mechanism SHOULD accept any SNMP-Set operation on the optional docsDevEvReporting bit-value and always report value zero for the optional bit on SNMP-Get operations.

Table 20. Default Event priorities for CMTS supporting both local-log non-volatile and local-log volatile[55]

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency	Yes	No	No	No or Yes*
2 Alert	Yes	No	No	No or Yes*
3 Critical	Yes	Yes	Yes	No or Yes*
4 Error	No or Yes**	Yes	Yes	Yes
5 Warning	No or Yes**	Yes	Yes	Yes
6 Notice	No or Yes**	Yes	Yes	Yes
7 Informational	No	No	No	No
8 Debug	No	No	No	No

*Note: CMTS/CM events designated for local-nonvolatile storage MAY also be retained in local-volatile storage

**Note: CMTS/CM events designated for local-volatile storage MAY also be retained in local-nonvolatile storage.

Notifications for standard DOCSIS events generated by the CMTS MUST be in the format specified in Appendix H.

4.4.2.5 Event Priorities for DOCSIS and Vendor Specific Events[56][57]

DOCSIS 1.1 compliant cable device (CMTS/CM) MUST strictly assign DOCSIS and Vendor specific events accordingly to Table-21.

Table 21. Event Priorities Assignment For CM and CMTSs

Event Priority	CM Event Assignment	CMTS Event Assignment
1 Emergency	Vendor Specific	Vendor Specific
2 Alert	DOCSIS and Vendor Specific (optional*)	Vendor Specific
3 Critical	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
4 Error	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional)
5 Warning	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional)
6 Notice	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional)
7 Informational	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
8 Debug	Vendor Specific	Vendor Specific

* Vendor-specific optional event definitions are recommended only where the CM/CMTS allows for sufficient storage of such events.

[55] Revised table and added two footnotes per ECN OSS-N-03006 by GO on 04/21/03.

[56] Added new Section 4.4.2.5 per ECN OSS-N-02192, chg #2, by GO, on 12/06/02.

[57] Revised Table 21 per ECN OSS-N-03045 by GO on 05/01/03.

4.4.3 Throttling, Limiting And Priority For Event, Trap and Syslog

4.4.3.1 Trap and Syslog Throttling, Trap and Syslog Limiting

DOCSIS 1.1 compliant cable device (CMTS/CM) MUST support SNMP TRAP/INFORM and SYSLOG throttling and limiting as described in RFC-2669, regardless of SNMP mode.

4.4.3.2 Maximum Priorities for Event Reporting

The Table 17 , Table 18, Table 19 and Table 20 in 4.4.2 define the default required event reporting capacity for events with different priorities for CM and CMTS. This capacity can be considered the minimum requirement for vendors to implement. Vendors may choose to report an event with more mechanisms than required in the tables. According to the priority definitions, there is a maximum level that an event can be reported. Table 22 shows that maximum level for CM events and Table 23 displays that for CMTS events.

The vendor-specific priorities can be handled differently by different vendors in their own ways.

Table 22. Maximum Level of Support for CM Events[58]

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency				
2 Alert	Yes			Yes
3 Critical	Yes			Yes
4 Error	Yes	Yes	Yes	Yes
5 Warning	Yes	Yes	Yes	Yes
6 Notice	Yes	Yes	Yes	Yes
7 Informational	Yes	Yes	Yes	Yes
8 Debug	Yes	Yes	Yes	Yes

Table 23. Maximum Level of Support for CMTS Events

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency				
2 Alert				
3 Critical	Yes	Yes	Yes	Yes
4 Error	Yes	Yes	Yes	Yes
5 Warning	Yes	Yes	Yes	Yes
6 Notice	Yes	Yes	Yes	Yes
7 Informational	Yes	Yes	Yes	Yes
8 Debug				

[58] Revised Table 22 per ECN OSS-N-03045 by GO on 05/01/03.

4.4.3.3 BIT Values for docsDevEvReporting (RFC-2669)

Permissible BITs values for RFC-2669 object docsDevEvReporting include:

- 1:local-nonvolatile(0)
- 2:traps(1)
- 3:syslog(2)
- 4:local-volatile(3)

An event reported by SNMP-Trap or SYSLOG MUST be accompanied by a Local-Log. The following BITS type values for RFC-2669 object docsDevEvReporting MUST NOT be accepted:

- 0x20 = syslog only
- 0x40 = trap only
- 0x60 = (trap + syslog) only

Note that the lower nibble MUST be zero in all cases, resulting in thirteen acceptable value s.

docsDevEvReporting SNMP SET requests for unacceptable values MUST result in a 'Wrong Value' error for SNMPv2c/v3 PDUs or a 'Bad Value' error for SNMPv1 PDUs.

When both local-log non-volatile and local-log volatile bits are set for a specific docsDevEvReporting event priority, the non-volatile storage MUST be maintained and the volatile storage MAY be maintained, since active functionality is identical. When both local-log non-volatile and local-log volatile bits are set for a specific docsDevEvReporting event priority, events MUST NOT be reported in duplicate through the docsDevEventTable.

4.4.4 Non-SNMP Fault Management Protocols

The OSS can use a variety of tools and techniques to examine faults at multiple layers. For the IP layer, useful non-SNMP based tools include ping (ICMP Echo and Echo Reply), traceroute (UDP and various ICMP Destination Unreachable flavors). Pings to a CM from its CMCI side MUST be supported to enable local connectivity testing from a customer's PC to the modem. The CM and CMTS MUST support IP end-station generation of ICMP error messages and processing of all ICMP messages.

4.5 Performance Management

At the CATV MAC and PHY layers, performance management focuses on the monitoring of the effectiveness of cable plant segmentation and rates of upstream traffic and collisions. Instrumentation is provided in the form of the standard interface statistics [RFC-2863], as well as the docsifCmtsServiceTable and docsifCmServiceTable entries. It is not anticipated that the CMTS upstream bandwidth allocation function will require active network management intervention and tuning.

At the LLC layer, the performance management focus is on bridge traffic management. The CM and CMTS (if the CMTS implements transparent bridging) MUST implement the Bridge MIB RFC-1493, including the dot1dBase and dot1dTp groups. The CM and CMTS MUST implement a managed object that controls whether the 802.1d spanning tree protocol (STP) is run and topology update messages are generated; STP is unnecessary in hierarchical, loop-free topologies. If the STP is enabled for the CM/CMTS, then the CM/CMTS MUST implement the dot1dStp group. These MIB groups' objects allow the NMS to detect when bridge forwarding tables are full, and enable the NMS to modify aging timers.

A final performance concern is the ability to diagnose unidirectional loss. Both the CM and CMTS MUST implement the mib-2 Interfaces group [RFC-2863].[59] When there exists more than one upstream or downstream channel, the CM/CMTS MUST implement an instance of IfEntry for each channel. The ifStack group [RFC-2863] MUST be used to define the relationships among the CATV MAC interfaces and their channels.

4.5.1 Additional MIB Implementation Requirements

To support performance monitoring and data collection for capacity, fault, and performance management, CM and CMTS MUST support MIB objects with:

- Accurate in measurement
- Counter properly working (i.e. counter roll over at maximum)
- Correct counter capacity
- Counter reset properly
- Update rate of 1 second

4.6 Coexistence

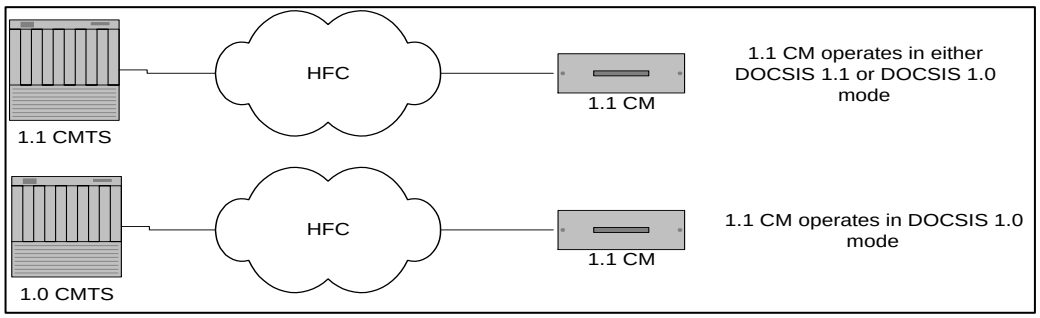

Figure 8. Coexistent (DOCSIS 1.0 mode VS DOCSIS 1.1 mode)

When DOCSIS 1.1 compliant CM is connected to 1.1 CMTS, it can operate in either DOCSIS 1.1 mode or DOCSIS 1.0 mode. When DOCSIS 1.1 compliant CM is connected to 1.0 CMTS, it operates in DOCSIS 1.0 mode. Refer to [DOCSIS 5] and BPI+ specifications for more detail descriptions of what features are available when DOCSIS 1.1 compliant CM is operating in different modes.

[59] Revised text per ECN OSS-N-03066 by GO on 0/10/03.

4.6.1 Coexistence and MIBs[60, 61]

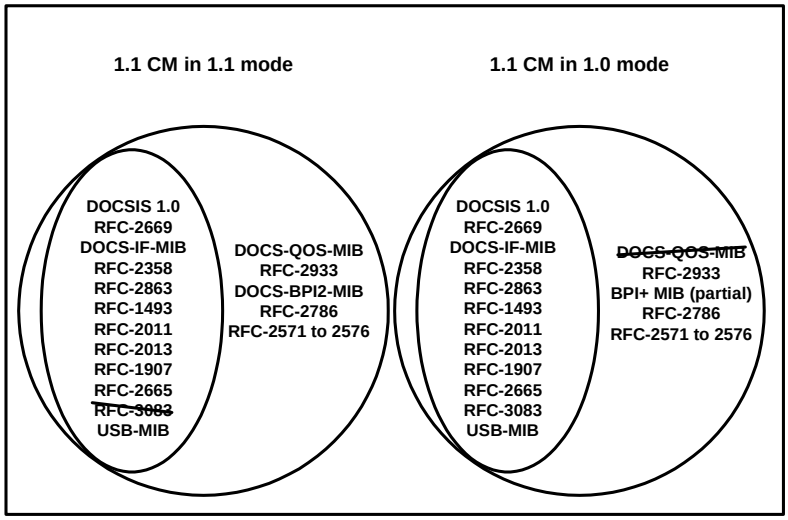

Figure 9. CM DOCSIS Mode and MIBs Requirement

4.6.1.1 Requirements for 1.1 CM operating in 1.1 mode

- RFC-2669
- DOCS-IF-MIB (certain objects are optional – refer to Appendix A)
- RFC-2665
- RFC-1493
- RFC-2011
- RFC-2013
- RFC-2933
- USB-MIB
- DOCS-QOS-MIB
- DOCS-CABLE-DEVICE-TRAP-MIB (see Appendix M)
- DOCS-BPI2-MIB
- RFC-2786 (When CM is in SNMP V1/V2c NmAccess mode, CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all the request to tables and objects in V3Kickstart.)
- RFC-3411 to 3415 (When CM is in SNMP v1/v2c NmAccess mode, CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all the request to tables and objects defined in RFC-3411 to 3415 and RFC-2576.)

When DOCSIS 1.1 compliant CM operates in 1.1 mode, it MUST NOT support the following MIB(s):

- RFC-3083

DOCS-BPI-MIB MUST not be available for any access from SNMP manager. DOCSIS 1.1 compliant CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all requests to tables and objects in DOCS-BPI-MIB.

[60] Updated and revised various text and reference statements per ECN OSS-N-03066 by GO on 07/10/03.

[61] Figure 9 updated per OSS-N-02190 by GO on 07/29/03.

4.6.1.2 Requirements for 1.1 CM operating in 1.0 mode[62]

When DOCSIS 1.1 compliant CM operates in 1.0 mode, it MUST support the following MIBs:

- RFC-2669
- DOCS-IF-MIB (certain objects are optional – refer to Appendix A)
- RFC-2665
- RFC-1493
- RFC-2011
- RFC-2013
- RFC-2933 (IF CM in 1.0 mode supports IGMP, it must implement RFC-2933)
- USB-MIB
- RFC-3083
- DOCS-BPI2-MIB. Part of the DOCS-BPI2-MIB MUST be supported. Refer to Appendix A for specific MIB object requirements.
- RFC-2786 (When CM is in SNMP V1/V2c NmAccess mode, CM MUST respond with "NoSuchName" or corresponding SNMPv2c error code "NoAccess" for all the request to tables and objects in V3Kickstart.)
- RFC-3411 to 3415 and RFC-2576 (When CM is in SNMP v1/v2c NmAccess mode, CM MUST respond with "NoSuchName" or corresponding SNMPv2c error code "NoAccess" for all the request to tables and objects defined in RFC-3411 to 3415 and RFC-2576.)

When DOCSIS 1.1 compliant CM operates in 1.0 mode, it MUST NOT support the following MIB(s):

- DOCS-QOS-MIB
- DOCS-BPI2-MIB (part of the BPI+ MIB MUST still be supported to enable secure software download. Refer to Appendix A for specific MIB object requirements.)
- DOCS-QOS-MIB and DOCS-BPI2-MIB, MUST not be available for any access from SNMP manager. DOCSIS 1.1 compliant CM MUST respond with "NoSuchName" or corresponding SNMPv2c error code "NoAccess" for all requests to tables and objects in DOCS-QOS-MIB and DOCS-BPI2-MIB.

When DOCSIS 1.1 CM operates at 1.0 mode, it MAY (optional) support DOCS_CABLE-DEVICE-TRAP-MIB. Some of the traps will not be applicable. See Appendix M.

4.6.2 Coexistence and SNMP

DOCSIS 1.1 compliant CM MUST support SNMPv3 and SNMPv1/v2c functionality as specified in Section 2 regardless of what mode (DOCSIS 1.0 or DOCSIS 1.1) CM operates in.

[62] Updated and revised various text and reference statements per ECN OSS-N-03066 by GO on 07/10/03.

5 OSS for BPI+

This section provides the requirements, guidelines, and/or examples related to the Digital Certificate management process and policy.

5.1 DOCSIS Root CA

The DOCSIS Root CA issues two kinds of the digital certificates as specified by the BPI+ specification. One is the Manufacturer CA Certificate embedded in the DOCSIS 1.1 compliant CM and verified by the CMTS in order to authenticate the CM during the CM initialization when the CM is provisioned to enable BPI+. The other is the Manufacturer Code Verification Certificate (CVC) embedded in the CM Code File and verified by the CM in order to authenticate the CM Code File during the Secure Software Downloading regardless of whether the BPI+ is provisioned or not.

The legitimate DOCSIS Root CA Certificate needs to be delivered to the cable operators and/or the CMTS vendors because the legitimate DOCSIS Root CA Certificate MUST be provisioned in the CMTS in order to realize the correct CM Authentication. The legitimate DOCSIS Root CA Certificate also needs to be delivered to the CM vendors because the legitimate DOCSIS Root CA Public Key extracted from the legitimate DOCSIS Root CA Certificate MUST be embedded in the CM in order for the CM to verify the CVC in the CM Code File. Since the DOCSIS Root CA Certificate is not a secret, the DOCSIS Root CA MAY disclose the DOCSIS Root CA Certificate to any organization including the cable operators, the CMTS vendors, and the CM vendors.

5.2 Digital Certificate Validity Period and Re-issuance

5.2.1 DOCSIS Root CA Certificate

The validity period of the DOCSIS Root CA Certificate is 30 years. The re-issuance process is TBD.

5.2.2 DOCSIS Manufacturer CA Certificate

When the DOCSIS Root CA newly issues the DOCSIS Manufacturer CA Certificate,

- the tbsCertificate.validity.notBefore MUST be the actual issuance date and time, and
- tbsCertificate.validity.notAfter MUST be the actual issuance date and time plus 20 years.

Before the DOCSIS Manufacturer CA Certificate expires, the certificate with the same information except the tbsCertificate.validity.notAfter and tbsCertificate.serialNumber needs to be re-issued. The DOCSIS 1.1 compliant CM vendors MUST obtain the re-issued DOCSIS Manufacturer CA Certificate from the DOCSIS Root CA at least two years before the tbsCertificate.validity.notAfter value of the current DOCSIS Manufacturer CA Certificate.

When the DOCSIS Root CA re-issues the DOCSIS Manufacturer CA Certificate, the following attribute values MUST be the same with the current DOCSIS Manufacturer CA Certificate:

- tbsCertificate.issuer
- tbsCertificate.subject
- tbsCertificate.subjectPublicKeyInfo

As well, the tbsCertificate.validity.notAfter MUST be the actual re-issuance date and time plus 20 years.

5.2.3 DOCSIS CM Certificate

The requirements for the DOCSIS CM Certificate including the validity period are specified by the BPI+ specification.

5.2.4 DOCSIS Code Verification Certificate

When the DOCSIS Root CA newly issues the DOCSIS Manufacturer Code Verification Certificate (CVC), the following conditions apply:

- the tbsCertificate.validity.notBefore MUST be the actual issuance date and time
- tbsCertificate.validity.notAfter MUST NOT exceed the actual issuance date and time by 10 years, and MUST be valid at least 2 years from the actual issuance date.[63]

Before the DOCSIS Manufacturer CVC expires, the certificate with the same information except the tbsCertificate.validity.notBefore, the tbsCertificate.validity.notAfter and tbsCertificate.serialNumber needs to be re-issued. The DOCSIS 1.1 compliant CM vendors MUST obtain the re-issued DOCSIS Manufacturer CVC from the DOCSIS Root CA at least 6 months before the tbsCertificate.validity.notAfter value of the current DOCSIS Manufacturer CVC.

: When the DOCSIS Root CA re-issues the DOCSIS Manufacturer CVC, the following attribute values MUST be the same as the current DOCSIS Manufacturer CVC:

- tbsCertificate.issuer
- tbsCertificate.subject[64]

As well, the tbsCertificate.validity.notBefore MUST be between the tbsCertificate.validity.notBefore value of the current DOCSIS Manufacturer CVC, and the actual issuance date and time. In addition, the tbsCertificate.validity.notAfter MUST be the actual re-issuance date and time plus 2 to 10 years.[65]

5.3 CM Code File Signing Policy

The CM vendor and the cable operator can control the Secure Software Download process based on their policy by updating the Manufacturer/Co-Signer CVC and/or by changing the signingTime in the Manufacturer/Co-Signer CVS (Code Verification Signature). At this time, the DOCSIS 1.1 specifications don't specify the policy related to the CM Code File signing process. However, an example of the policy is specified in this section.

5.3.1 Manufacturer CM Code File Signing Policy

The DOCSIS 1.1 compliant CM vendor and its Manufacturer Code Signing Agent (Mfg CSA), which securely stores the RSA private key corresponding to the RSA public key in the Manufacturer CVC and generates the CVS for the CM Code File, MAY employ the following policy for the CM Code File signing process.

The Mfg CSA continues to put the exact same date and time value (T1) in the signingTime field in the Mfg CVS of the CM Code File as long as the vendor does not have any CM Code File to revoke.

---

[63] Revised bulleted statement per ECN OSS-N-02204 by GO on 11/12/02.

[64] Revised the preceeding paragraph and bulleted list per ECN OSS-N-03054 by GO on 06/04/03.

[65] Revised paragraph per ECN OSS-N-02204 by GO on 11/12/02.

Once the vendor realizes the certain issues in one or more CM Code File(s) and wants to revoke them, the vendor choose the current date and time value (T2) and starts using T2 as the signingTime value in the Mfg CVS for all the newly created CM Code File from that point. In addition, re-sign all the good old CM Code Files using the T2.

Under this policy, because the multiple CM Code Files make a group of the CM Code Files with the exact same signingTime value in the Msg CVS, the operator can download any CM Code File in the group in any order. That is, among the CM Code Files in the same group, the software downgrade can be realized.

6 OSSI for CMCI

This section defines the operational mechanisms needed to support the transmission of data over cable services between a cable modem and the customer premise equipment. More specifically, this section will outline the following:

- SNMP access via CMCI
- Console Access
- CM diagnostic capabilities
- Protocol Filtering
- Required MIBs

Currently, the CMCI is categorized as internal, external, and CPE Controlled cable modem functional reference models. The external cable modems MAY have either an Ethernet 10BASE-T or Universal Serial Bus (USB) CMCI interface or both. If both interfaces are present on a CM, they MAY be active at the same time.

The internal cable modems MUST utilize the Peripheral Component Interface (PCI) bus for transparent bi-directional IP traffic forwarding. The PCI interface MUST be defined and accessible from an SNMP manager for both operational and security purposes.

The CPE Controlled Cable modems (CCCM) CMCI MAY be either a Peripheral Component Interface (PCI) or Universal Serial Bus (USB) interface. If PCI is utilized, the interface MUST be defined and accessible from an SNMP manager for both operational and security purposes.

6.1 SNMP Access via CMCI

SNMP access from the CMCI before and after completing the CMTS registration process, MUST comply with the access requirements specified in section 2.2. The CM MUST support SNMP access through the following IP addresses:

1) The CM DHCP-acquired IP MUST accept an SNMP request from CMCI only after completing registration.

2) The CM MUST support 192.168.100.1 as the well-known diagnostic IP address accessible only from the CMCI interfaces regardless of the CM registration state. The well-known diagnostic IP address, 192.168.100.1, MUST be supported on all physical interfaces associated with the CMCI (e.g. USB, 10Base-T, etc.). SNMP requests coming from the CATV interface targeting the well-known IP MUST be dropped by the CM.

CM MAY also implement alternative interfaces like link-local method described in the IETF document “draft-ietf-zeroconf-ipv4-linklocal-05.” [IETF10]. If implemented, the CM MUST restrict the IP address range described in “Ipv4 Link-local address selection, defense and delivery” of the mentioned document to 169.254.1.0 – 169.254.1.255.”

6.2 Console Access

An external cable modem MUST NOT allow access to the CM functions via a console port. For this specification, a console port is defined as a communication path, either hardware or software that allows a user to issue commands to modify the configuration or operational status of the CM. Access to the external CM MUST only be allowed using DOCSIS 1.1 defined RF interfaces and operator-controlled SNMP access via the CMCI.

6.3 CM Diagnostic Capabilities

The CM MAY have a diagnostic interface for debugging and troubleshooting purposes. The interface MUST be limited by default to the requirements described in Section 2.2, part (a) before and after registration, and SHOULD be disabled by default after registration has been completed. Additional controls MAY be provided that will enable the MSO to alter or customize the diagnostic interface, such as via the configuration process or later management by the MSO through the setting of a proprietary mib.

6.4 Protocol Filtering

The CM MUST be capable of filtering all broadcast traffic from the host CPE, with the exception of DHCP and ARP packets. This filtering function must adhere to section 4.3 (Protocol Filters) of this document. All ICMP type packets MUST be forwarded from the CMCI interface to the RF upstream interface. The CMCI MUST also adhere to the data forwarding rules defined in [DOCSIS 5].

6.5 Management Information Base (MIB) Requirements

All Cable Modems MUST implement the MIBs detailed in section 3 (Management Information Bases) of this specification, with the following exceptions:

- An external CM with only USB interface(s), MUST NOT implement RFC-2665: Ethernet Interface MIB.
- An external CM with only USB interface(s), MUST implement the IETF Proposed Standard RFC version of USB-MIB.
- An internal CM MAY implement RFC-2665: Interface MIB.

7 CM Operational Status Visualization[66]

DOCSIS 1.1 compliant CM is RECOMMENDED to support a standard front-panel LEDs (Light Emitting Diode) that presents straightforward information about the registration state of the CM to facilitate efficient customer support operations.

7.1 CM LEDs Requirements and Operation

The LEDs on a DOCSIS 1.1 compliant CM SHOULD have three states; 1) unlit, 2) flash, 3) lit solid. A 'flash' LED SHOULD turn on and off with a 50% duty cycle at a frequency not less than 2 cycles per second.

The LEDs will light sequentially, following the normal CM boot-up procedure as specified in the DOCSIS RFI specification. In this way, the installer can detect a failure that prevents the CM from becoming operational.

DOCSIS 1.1 compliant CMs is RECOMMENDED to have a minimum of five LEDs visible on the outside case divided in three functional groups:

- BOX: It SHOULD have 1 LED labeled as POWER
- DOCSIS: This group has LEDs for the DOCSIS interface. It SHOULD have 3 LEDs labeled as DS, US, ONLINE
- CPE: This Group has the LINK LED indication. It SHOULD have a minimum of 1 LED labeled as LINK. DOCSIS 1.1 CMs MAY have multiple LEDs in the CPE Group to represent individual CPE interfaces types and parameters. Those LEDs MAY be labeled accordingly to their associated interface type.

There is no specific requirement for labeling the functional groups, moreover, the LEDs in the DOCSIS group SHOULD be in the order DS, US, ONLINE from left to right or Top to Bottom, as appropriate for the orientation of the device. As well, the overall LED distribution SHOULD intent to be in the order POWER, DS, US, ONLINE, LINK.

The RECOMMENDED LEDs indicate the following steps are in progress or have completed successfully by the CM:

- Power on and optionally any proprietary CM self-test
- DOCSIS Downstream Scanning and Sync
- DOCSIS Upstream Channel Selection and Ranging
- Becoming operational
- Data Link and Activity

[66] Added Section 7 per ECN OSS-N-03024

NOTE: The RECOMMENDED LEDs SHOULD operate as described below:

7.1.1 Power and self test

When the CM is turned on, the RECOMMENDED LEDs, or at least the DOCSIS Group LEDs (DS, US, ONLINE), SHOULD ‘flash’ while the CM performs the system initialization of the Operational System, CM application load, and any proprietary self-tests. Following the successful completion of the steps above, the RECOMMENDED LEDs, or at least the DOCSIS Group LEDs, SHOULD show “lit solid” for one second, and then only the POWER LED SHOULD remain ‘lit solid’. The LINK LED MAY also be ‘lit solid’ if a CPE device is properly connected (see 7.1.5 below). If the system initialization described above results in a failure, the RECOMMENDED LEDs, or at least the DOCSIS Group LEDs, SHOULD continue to ‘flash’.

7.1.2 Scanning and Synchronization to Downstream

DS: The DS LED SHOULD ‘flash’ as the CM scans for a Downstream DOCSIS channel. The DS LED SHOULD go to ‘lit solid’ when the CM MAC layer has already synchronized, as defined in [DOCSIS 5], section 9.2.1. Whenever the CM is scanning for a downstream channel and attempting to synchronize to a downstream channel, the DS LED SHOULD ‘flash’ and the US and ONLINE LEDs SHOULD be ‘unlit’.

7.1.3 DOCSIS Upstream obtaining parameters

US: After the DS LED goes ‘lit solid’, the US LED SHOULD 'flash', and the ONLINE LED SHOULD be ‘unlit' while the CM is obtaining upstream parameters and performing initial ranging. When the CM Completes a successful initial Ranging, the US LED SHOULD go 'lit solid' (See Figure 9-3 Obtaining US parameters [DOCSIS 5]).

7.1.4 Becoming Operational

ONLINE: After the US LED goes ‘lit solid’, the ONLINE LED SHOULD ‘flash’, while the CM continues the process to become operational. When the CM is operational, the ONLINE LED SHOULD be 'lit solid'. Operational is defined according to [DOCSIS5], Figure 9-1, CM initialization overview. If at any point there is a failure in the registration process that causes the CM to not become operational (ranging, DHCP, configuration file download, registration, Baseline Privacy initialization, etc.), the ONLINE LED SHOULD continue to 'flash'.

If the CM becomes operational and the CM configuration file has the Network Access Control Object (NACO) set to off, the ONLINE LED SHOULD be ‘unlit’, while ‘DS and US LEDS SHOULD 'flash' .

7.1.5 Data Link and Activity

LINK ACTIVITY: This LED SHOULD be 'lit solid' when a CPE device is connected and the CM is not bridging data. The LED SHOULD only ‘flash’ when the CM is bridging data during the CM operational state and NACO=1. The Link LED SHOULD not ‘flash’ for data traffic originating or terminating at the CM device itself.

If link is detected with a CPE device, the LINK LED MAY ‘lit solid’ any time after Power and self test step is completed.

7.2 Additional CM Operational Status Visualization Features

It is acceptable to change the DOCSIS defined LED behavior when the CM is in a vendor proprietary mode of operation. A DOCSIS 1.1 Compliant CM MUST NOT have additional LEDs that reveal DOCSIS specific information about the configuration file content, or otherwise clearly specified (see NACO visualization in section 7.1.4 and 7.1.5).

7.2.1 Software Download

The CM Should signal that a Software Download [DOCSIS 6] Appendix D is in process by indicating DS and US LEDs to ‘flash’ and ONLINE LED ‘lit solid’.

Appendix A. Detailed MIB Requirements

NOTE:

ACC-FN- Accessible for Notify
D - Deprecated
M - Mandatory
N-Acc - Not accessible
NA - Not Applicable
N-Sup - MUST not support
O - Optional
Ob - Obsolete
RC - Read-Create
RO - Read-Only
RW - Read-Write
RC/RO – Read-Create or Read-Only
RW/RO – Read-Write or Read-Only

General rules:

D - Deprecated – It is optional. That is, a vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object MUST be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

M - Mandatory – The object MUST be implemented correctly according to the MIB definition.

N-Acc - Not Accessible – The object is not accessible and is usually an index in a table.

NA - Not Applicable – Not applicable to the device.

N-Sup - MUST Not Support – Device MUST NOT support the object. That is, an agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

O - Optional – A vendor can choose to implement or not implement the object. If a vendor chooses to implement the object, the object MUST be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, an agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

Ob - Obsolete – It is optional. Though in SNMP convention, obsolete objects should not be implemented, DOCSIS 1.1 OSSI lets vendors choose whether or not to support the obsolete object. That is, a vendor can choose to implement or not implement the obsolete object. If a vendor chooses to implement the object, the object MUST be implemented correctly according to the MIB definition. If a vendor chooses not to implement the object, SNMP agent MUST NOT instantiated such object and MUST respond with the appropriate error/exception condition. (e.g., no such object for SNMPv2c)

RC – Read-Create – The access of the object MUST be implemented as Read-Create.

RO – Read-Only – The access of the object MUST be implemented as Read-Only.

RW – Read-Write – The access of the object MUST be implemented as Read-Write.

RC/RO – Read-Create or Read-Only – The access of the object MUST be implemented as either Read-Create or Read-Only as described in the MIB definition.

RW/RO – Read-Write or Read-Only – The access of the object MUST be implemented as either Read-Write or Read-Only as described in the MIB definition.

ATRAP – Accessible through SNMP trap

Table 24. Detailed MIB Requirements[67, 68, 69, 70, 71, 72, 73]

DOCS-IF-MIB				
docsIfDownstreamChannelTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfDownChannelId	M	RO	M	RO
docsIfDownChannelFrequency	M	RO	M	RW/RO
docsIfDownChannelWidth	M	RO	M	RW/RO
docsIfDownChannelModulation	M	RO	M	RW
docsIfDownChannelInterleave	M	RO	M	RW
docsIfDownChannelPower	M	RO	M	RW/RO
docsIfDownChannelAnnex	O	RO	O	RW/RO
docsIfUpstreamChannelTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfUpChannelId	M	RO	M	RO
docsIfUpChannelFrequency	M	RO	M	RW
docsIfUpChannelWidth	M	RO	M	RW
docsIfUpChannelModulationProfile	M	RO	M	RW
docsIfUpChannelSlotSize	M	RO	M	RW/RO
docsIfUpChannelTxTimingOffset	M	RO	M	RO
docsIfUpChannelRangingBackoffStart	M	RO	M	RW
docsIfUpChannelRangingBackoffEnd	M	RO	M	RW
docsIfUpChannelTxBackoffStart	M	RO	M	RW
docsIfUpChannelTxBackoffEnd	M	RO	M	RW
docsIfUpChannelScdmaActiveCodes	O	RO	O	RC
docsIfUpChannelScdmaCodesPerSlot	O	RO	O	RC
docsIfUpChannelScdmaFrameSize	O	RO	O	RC
docsIfUpChannelScdmaHoppingSeed	O	RO	O	RC
docsIfUpChannelType	O	RO	O	RC
docsIfUpChannelCloneFrom	O	RO	O	RC
docsIfUpChannelUpdate	O	RO	O	RC
docsIfUpChannelStatus	O	RO	O	RC

[67] Account Management MIB deleted from Table 23 per OSS-N-02136 by RKV on 10/23/02. DOCS-CABLE-DEVICE-MIB (RFC 2669) updated per OSS-N-02167 by RKV on 10/24/02.

[68] Changed “BPI+MIB (draft-ietf-ipcdn-bpiplus-mib-07.txt” to “BPI+MIB (draft-ietf-ipcdn-bpiplus-mib-05.txt” per ECN OSS-N-03020 (rescinds OSS-N-02229) by GO on 03/21/03.

[69] Added note to “SNMP Management Framework architecture (RFC-2571)” per ECN OSS-N-03013 by GO on 02/25/03

[70] Revised Table 24 per ECN OSS-N-03022 by GO on 03/21/03.

[71] Revised Table 24 per ECN OSS-N-03066 by GO on 07/10/03.

[72] Revised Table 24 per ECN OSS-N-03068 by GO on 07/11/03.

[73] Revised Table 24 per ECN OSS-N-03070 by GO on 07/11/03.

Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsIfUpChannelPreEqEnable	O	RO	M	RO	M	RW/RC
docsIfQosProfileTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**Object**	**1.1CM in 1.0 mode**
docsIfQosProfiIndex	M	N-Acc	O	N-Acc		M
docsIfQosProfPriority	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxUpBandwidth	M	RO	O	RO	O	RC/RO
docsIfQosProfGuarUpBandwidth	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxDownBandwidth	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxTxBurst	D	RO	D	RO	D	RC/RO
docsIfQosProfBaselinePrivacy	M	RO	O	RO	O	RC/RO
docsIfQosProfStatus	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxTransmitBurst	M	RO	O	RO	O	RC/RO
docsIfSignalQualityTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfSigQIncludesContention			M	RO	M	RO
docsIfSigQUnerroreds			M	RO	M	RO
docsIfSigQCorrecteds			M	RO	M	RO
docsIfSigQUncorrectables			M	RO	M	RO
docsIfSigQSignalNoise			M	RO	M	RO
docsIfSigQMicroreflections			M	RO	M	RO
docsIfSigQEqualizationData			M	RO	M	RO
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**Object**	**1.1CM in 1.0 mode**
docsIfSigQExtUnerroreds	O	RO	O	RO	M	RO
docsIfSigQExtCorrecteds	O	RO	O	RO	M	RO
docsIfSigQExtUncorrectables	O	RO	O	RO	M	RO
docsIfCmMacTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmCmtsAddress			M	RO	NA	NA
docsIfCmCapabilities			M	RO	NA	NA
docsIfCmRangingRespTimeout			Ob	N-Sup	NA	NA
docsIfCmRangingTimeout			M	RW	NA	NA
docsIfCmStatusTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmStatusValue			M	RO	NA	NA
docsIfCmStatusCode			M	RO	NA	NA
docsIfCmStatusTxPower			M	RO	NA	NA

docsIfCmStatusResets			M	RO	NA	NA
docsIfCmStatusLostSyncs			M	RO	NA	NA
docsIfCmStatusInvalidMaps			M	RO	NA	NA
docsIfCmStatusInvalidUcds			M	RO	NA	NA
docsIfCmStatusInvalidRangingResponses			M	RO	NA	NA
docsIfCmStatusInvalidRegistrationResponses			M	RO	NA	NA
docsIfCmStatusT1Timeouts			M	RO	NA	NA
docsIfCmStatusT2Timeouts			M	RO	NA	NA
docsIfCmStatusT3Timeouts			M	RO	NA	NA
docsIfCmStatusT4Timeouts			M	RO	NA	NA
docsIfCmStatusRangingAborteds			M	RO	NA	NA
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsIfCmStatusDocsisOperMode	O	RO	M	RO	NA	NA
docsIfCmStatusModulationType	O	RO	M	RO	NA	NA
docsIfCmStatusEqualizationData	O	RO	M	RO	NA	NA
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsChannelUtilizationInterval			NA	NA	M	RW
DocsIfCmtsChannelUtilizationTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsChannelUtIfType			NA	NA	M	N-Acc
docsIfCmtsChannelUtId			NA	NA	M	N-Acc
docsIfCmtsChannelUtUtilization			NA	NA	M	RO
DocsIfCmtsDownChannelCounterTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsDownChnlCtrId			NA	NA	M	RO
docsIfCmtsDownChnlCtrTotalBytes			NA	NA	M	RO
docsIfCmtsDownChnlUsedBytes			NA	NA	M	RO
docsIfCmtsDownChnlExtTotalBytes			NA	NA	M	RO
docsIfCmtsDownChnlExtUsedBytes			NA	NA	M	RO
DocsIfCmtsUpChannelCounterTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsUpChnlCtrId			NA	NA	M	RO
docsIfCmtsUpChnlCtrTotalMslots			NA	NA	M	RO
docsIfCmtsUpChnlCtrUcastGrantedMslots			NA	NA	M	RO
docsIfCmtsUpChnlCtrTotalCntnMslots			NA	NA	M	RO
docsIfCmtsUpChnlCtrUsedCntnMslots			NA	NA	M	RO

docsIfCmtsUpChnlCtrExtTotalMslots	NA	NA	M	RO
docsIfCmtsUpChnlCtrExtUcastGrantedMslots	NA	NA	M	RO
docsIfCmtsUpChnlCtrExtTotalCntnMslots	NA	NA	M	RO
docsIfCmtsUpChnlCtrExtUsedCntnMslots	NA	NA	M	RO
docsIfCmtsUpChnlCtrCollCntnMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrTotalCntnReqMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrUsedCntnReqMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrCollCntnReqMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrTotalCntnReqDataMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrUsedCntnReqDataMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrCollCntnReqDataMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrTotalCntnInitMaintMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrUsedCntnInitMaintMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrCollCntnInitMaintMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtCollCntnMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtTotalCntnReqMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtUsedCntnReqMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtCollCntnReqMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtTotalCntnReqDataMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtUsedCntnReqDataMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtCollCntnReqDataMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtTotalCntnInitMaintMslots	NA	NA	O	RO
docsIfCmtsUpChnlCtrExtUsedCntnInitMaintMslots	NA	NA	O	RO
docsIfCmServiceTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfCmServiceId	M	N-Acc	NA	NA
docsIfCmServiceQosProfile	M	RO	NA	NA
docsIfCmServiceTxSlotsImmed	M	RO	NA	NA
docsIfCmServiceTxSlotsDed	M	RO	NA	NA
docsIfCmServiceTxRetries	M	RO	NA	NA
docsIfCmServiceTxExceeds	M	RO	NA	NA
docsIfCmServiceRqRetries	M	RO	NA	NA
docsIfCmServiceRqExceeds	M	RO	NA	NA
docsIfCmServiceExtTxSlotsImmed	O	RO	NA	NA
docsIfCmServiceExtTxSlotsDed	O	RO	NA	NA
docsIfCmtsMacTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsCapabilities	NA	NA	M	RO
docsIfCmtsSyncInterval	NA	NA	M	RW/RO
docsIfCmtsUcdInterval	NA	NA	M	RW/RO
docsIfCmtsMaxServiceIds	NA	NA	M	RO
docsIfCmtsInsertionInterval	NA	NA	Ob	N-Sup

docsIfCmtsInvitedRangingAttempts	NA	NA	M	RW/RO
docsIfCmtsInsertInterval	NA	NA	M	RW/RO
docsIfCmtsStatusTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsStatusInvalidRangeReqs	NA	NA	M	RO
docsIfCmtsStatusRangingAborteds	NA	NA	M	RO
docsIfCmtsStatusInvalidRegReqs	NA	NA	M	RO
docsIfCmtsStatusFailedRegReqs	NA	NA	M	RO
docsIfCmtsStatusInvalidDataReqs	NA	NA	M	RO
docsIfCmtsStatusT5Timeouts	NA	NA	M	RO
docsIfCmtsCmStatusTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsCmStatusIndex	NA	NA	M	N-Acc
docsIfCmtsCmStatusMacAddress	NA	NA	M	RO
docsIfCmtsCmStatusIpAddress	NA	NA	M	RO
docsIfCmtsCmStatusDownChannelIfIndex	NA	NA	M	RO
docsIfCmtsCmStatusUpChannelIfIndex	NA	NA	M	RO
docsIfCmtsCmStatusRxPower	NA	NA	M	RO
docsIfCmtsCmStatusTimingOffset	NA	NA	M	RO
docsIfCmtsCmStatusEqualizationData	NA	NA	M	RO
docsIfCmtsCmStatusValue	NA	NA	M	RO
docsIfCmtsCmStatusUnerroreds	NA	NA	M	RO
docsIfCmtsCmStatusCorrecteds	NA	NA	M	RO
docsIfCmtsCmStatusUncorrectables	NA	NA	M	RO
docsIfCmtsCmStatusSignalNoise	NA	NA	M	RO
docsIfCmtsCmStatusMicroreflections	NA	NA	M	RO
docsIfCmtsCmStatusExtUnerroreds	NA	NA	O	RO
docsIfCmtsCmStatusExtCorrecteds	NA	NA	O	RO
docsIfCmtsCmStatusExtUncorrectables	NA	NA	O	RO
docsIfCmtsCmStatusDocsisRegMode	NA	NA	M	RO
docsIfCmtsCmStatusModulationType	NA	NA	M	RO
docsIfCmtsCmStatusInetAddressType	NA	NA	O	RO
docsIfCmtsCmStatusInetAddress	NA	NA	O	RO
docsIfCmtsCmStatusValueLastUpdate	NA	NA	O	RO
docsIfCmtsServiceTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsServiceId	NA	NA	M	N-Acc
docsIfCmtsServiceCmStatusIndex	NA	NA	D	RO
docsIfCmtsServiceAdminStatus	NA	NA	M	RW/RO
docsIfCmtsServiceQosProfile	NA	NA	M	RO
docsIfCmtsServiceCreateTime	NA	NA	M	RO
docsIfCmtsServiceInOctets	NA	NA	M	RO
docsIfCmtsServiceInPackets	NA	NA	M	RO
docsIfCmtsServiceNewCmStatusIndex	NA	NA	M	RO

docsIfCmtsModulationTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsModIndex			NA	NA	M	N-Acc
docsIfCmtsModIntervalUsageCode			NA	NA	M	N-Acc
docsIfCmtsModControl			NA	NA	M	RC
docsIfCmtsModType			NA	NA	M	RC
docsIfCmtsModPreambleLen			NA	NA	M	RC
docsIfCmtsModDifferentialEncoding			NA	NA	M	RC
docsIfCmtsModFECErrorCorrection			NA	NA	M	RC
docsIfCmtsModFECCodewordLength			NA	NA	M	RC
docsIfCmtsModScramblerSeed			NA	NA	M	RC
docsIfCmtsModMaxBurstSize			NA	NA	M	RC
docsIfCmtsModGuardTimeSize			NA	NA	M	RO
docsIfCmtsModLastCodewordShortened			NA	NA	M	RC
docsIfCmtsModScrambler			NA	NA	M	RC
docsIfCmtsModByteInterleaverDepth			NA	NA	O	RC
docsIfCmtsModByteInterleaverBlockSize			NA	NA	O	RC
docsIfCmtsModPreambleType			NA	NA	O	RC
docsIfCmtsModTcmErrorCorrectionOn			NA	NA	O	RC
docsIfCmtsModScdmaInterleaverStepSize			NA	NA	O	RC
docsIfCmtsModScdmaSpreaderEnable			NA	NA	O	RO
docsIfCmtsModScdmaSubframeCode			NA	NA	O	RC
docsIfCmtsModChannelType			NA	NA	O	RC
Object						
docsIfCmtsQosProfilePermissions			NA	NA	M	RW /RO
docsIfCmtsMacToCmTable						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsCmMac			NA	NA	M	N-Acc
docsIfCmtsCmPtr			NA	NA	M	RO
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsIfDocsisBaseCapability	O	RO	M	RO	M	RO
IF-MIB (RFC-2863)						
Object			**CM**	**Access**	**CMTS**	**Access**
ifNumber			M	RO	M	RO
IfTableLastChange			M	RO	M	RO
ifTable						

Object	**CM**	**Access**	**CMTS**	**Access**
ifIndex	M	RO	M	RO
ifDescr	M	RO	M	RO
ifType	M	RO	M	RO
ifMtu	M	RO	M	RO
ifSpeed	M	RO	M	RO
ifPhysAddress	M	RO	M	RO
ifAdminStatus	M	RW	M	RW
ifOperStatus	M	RO	M	RO
ifLastChange	M	RO	M	RO
ifInOctets	M	RO	M	RO
ifInUcastPkts	M	RO	M	RO
ifInNUcastPkts	D	RO	D	RO
ifInDiscards	M	RO	M	RO
ifInErrors	M	RO	M	RO
ifInUnknownProtos	M	RO	M	RO
ifOutOctets	M	RO	M	RO
ifOutUcastPkts	M	RO	M	RO
ifOutNUcastPkts	D	RO	D	RO
ifOutDiscards	M	RO	M	RO
ifOutErrors	M	RO	M	RO
ifOutQLen	D	RO	D	RO
ifSpecific	D	RO	D	RO
ifXTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
ifName	M	RO	M	RO
ifInMulticastPkts	M	RO	M	RO
ifInBroadcastPkts	M	RO	M	RO
ifOutMulticastPkts	M	RO	M	RO
ifOutBroadcastPkts	M	RO	M	RO
ifHCInOctets	O	RO	O	RO
ifHCInUcastPkts	O	RO	O	RO
ifHCInMulticastPkts	O	RO	O	RO
ifHCInBroadcastPkts	O	RO	O	RO
ifHCOutOctets	O	RO	O	RO
ifHCOutUcastPkts	O	RO	O	RO
ifHCOutMulticastPkts	O	RO	O	RO
ifHCOutBroadcastPkts	O	RO	O	RO
ifLinkUpDownTrapEnable	M	RW	M	RW
ifHighSpeed	M	RO	M	RO
ifPromiscuousMode	M	RW/RO	M	RW/RO
ifConnectorPresent	M	RO	M	RO
ifAlias	M	RW/RO	M	RW/RO
ifCounterDiscontinuityTime	M	RO	M	RO

ifStackTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
ifStackHigherLayer	M	N-Acc	M	N-Acc
ifStackLowerLayer	M	N-Acc	M	N-Acc
ifStackStatus	M	RC/RO	M	RC/RO
Object	**CM**	**Access**	**CMTS**	**Access**
ifStackLastChange	O	RO	O	RO
ifRcvAddressTable				
Object	**CM**	**Access**	**CMTS**	**Access**
ifRcvAddressAddress	O	N-Acc	O	N-Acc
ifRcvAddressStatus	O	RC	O	RC
IfRcvAddressType	O	RC	O	RC
Notification				
linkUp	M		M	
linkDown	M		M	
ifTestTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
ifTestId	O	RW	O	RW
ifTestStatus	O	RW	O	RW
ifTestType	O	RW	O	RW
ifTestResult	O	RO	O	RO
ifTestCode	O	RO	O	RO
ifTestOwner	O	RW	O	RW
BRIDGE-MIB (RFC-1493)				
NOTE: Implementation of BRIDGE MIB is required ONLY if device is a bridging device				
dot1dBase group				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dBaseBridgeAddress	M	RO	M	RO
dot1dBaseNumPorts	M	RO	M	RO
dot1dBaseType	M	RO	M	RO
dot1dBasePortTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dBasePort	M	RO	M	RO
dot1dBasePortIfIndex	M	RO	M	RO
dot1dBasePortCircuit	M	RO	M	RO
dot1dBasePortDelayExceededDiscards	M	RO	M	RO
dot1dBasePortMtuExceededDiscards	M	RO	M	RO
dot1dStp group				

NOTE: This group is required ONLY if STP is implemented				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dStpProtocolSpecification	M	RO	M	RO
dot1dStpPriority	M	RW	M	RW
dot1dStpTimeSinceTopologyChange	M	RO	M	RO
dot1dStpTopChanges	M	RO	M	RO
dot1dStpDesignatedRoot	M	RO	M	RO
dot1dStpRootCost	M	RO	M	RO
dot1dStpRootPort	M	RO	M	RO
dot1dStpMaxAge	M	RO	M	RO
dot1dStpHelloTime	M	RO	M	RO
dot1dStpHoldTime	M	RO	M	RO
dot1dStpForwardDelay	M	RO	M	RO
dot1dStpBridgeMaxAge	M	RW	M	RW
dot1dStpBridgeHelloTime	M	RW	M	RW
dot1dStpBridgeForwardDelay	M	RW	M	RW
dot1dStpPortTable				
NOTE: This table is required ONLY if STP is implemented				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dStpPort	M	RO	M	RO
dot1dStpPortPriority	M	RW	M	RW
dot1dStpPortState	M	RO	M	RO
dot1dStpPortEnable	M	RW	M	RW
dot1dStpPortPathCost	M	RW	M	RW
dot1dStpPortDesignatedRoot	M	RO	M	RO
dot1dStpPortDesignatedCost	M	RO	M	RO
dot1dStpPortDesignatedBridge	M	RO	M	RO
dot1dStpPortDesignatedPort	M	RO	M	RO
dot1dStpPortForwardTransitions	M	RO	M	RO
dot1dTp group				
Note: This group is required ONLY if transparent bridging is implemented.				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dTpLearnedEntryDiscards	M	RO	M	RO
dot1dTpAgingTime	M	RW	M	RW
dot1dTpFdbTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dTpFdbAddress	M	RO	M	RO
dot1dTpFdbPort	M	RO	M	RO
dot1dTpFdbStatus	M	RO	M	RO
dot1dTpPortTable				

Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dTpPort	M	RO	M	RO
dot1dTpPortMaxInfo	M	RO	M	RO
dot1dTpPortInFrames	M	RO	M	RO
dot1dTpPortOutFrames	M	RO	M	RO
dot1dTpPortInDiscards	M	RO	M	RO
dot1dStaticTable				
Note: Implementation of dot1dStaticTable is OPTIONAL				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot1dStaticAddress	O	RW	O	RW
dot1dStaticReceivePort	O	RW	O	RW
dot1dStaticAllowedToGoTo	O	RW	O	RW
dot1dStaticStatus	O	RW	O	RW
DOCS-CABLE-DEVICE-MIB (RFC-2669)				
docsDevBaseGroup				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevRole	M	RO	O	RO
docsDevDateTime	M	RO/RW	M	RW
docsDevResetNow	M	RW	O	RW
docsDevSerialNumber	M	RO	O	RO
docsDevSTPControl	M	RW/RO	O	RW/RO
docsDevNmAccessGroup				
NOTE: docsDevNmAccessGroup is NOT accessible when the device is in SNMP Coexistence mode.				
docsDevNmAcessTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevNmAccessIndex	M	N-Acc	O	N-Acc
docsDevNmAccessIp	M	RC	O	RC
docsDevNmAccessIpMask	M	RC	O	RC
docsDevNmAccessCommunity	M	RC	O	RC
docsDevNmAccessControl	M	RC	O	RC
docsDevNmAccessInterfaces	M	RC	O	RC
docsDevNmAccessStatus	M	RC	O	RC
docsDevNmAccessTrapVersion (Note: This object is currently not in RFC-2669)	M	RC	O	RC
docsDevSoftwareGroup				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevSwServer	M	RW	O	RW
docsDevSwFilename	M	RW	O	RW
docsDevSwAdminStatus	M	RW	O	RW
docsDevSwOperStatus	M	RO	O	RO
docsDevSwCurrentVers	M	RO	O	RO

docsDevServerGroup				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevServerBootState	M	RO	N-Sup	
docsDevServerDhcp	M	RO	N-Sup	
docsDevServerTime	M	RO	N-Sup	
docsDevServerTftp	M	RO	N-Sup	
docsDevServerConfigFile	M	RO	N-Sup	
docsDevEventGroup				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevEvControl	M	RW	M	RW
docsDevEvSyslog	M	RW	M	RW
docsDevEvThrottleAdminStatus	M	RW	M	RW
docsDevEvThrottleInhibited	M	RO	M	RO
docsDevEvThrottleThreshold	M	RW	M	RW
docsDevEvThrottleInterval	M	RW	M	RW
docsDevEvControlTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevEvPriority	M	N-Acc	M	N-Acc
docsDevEvReporting (Mandatory RW by DOCSIS 1.1; exception to RFC-2669)	M	RW	M	RW
docsDevEventTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevEvIndex	M	N-Acc	M	N-Acc
docsDevEvFirstTime	M	RO	M	RO
docsDevEvLastTime	M	RO	M	RO
docsDevEvCounts	M	RO	M	RO
docsDevEvLevel	M	RO	M	RO
docsDevEvId	M	RO	M	RO
docsDevEvText	M	RO	M	RO
docsDevFilterGroup				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevFilterLLCUnmatchedAction	M	RW	O	RW
docsDevFilterLLCTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevFilterLLCIndex	M	N-Acc	O	N-Acc
docsDevFilterLLCStatus	M	RC	O	RC
docsDevFilterLLCIfIndex	M	RC	O	RC
docsDevFilterLLCProtocolType	M	RC	O	RC
docsDevFilterLLCProtocol	M	RC	O	RC
docsDevFilterLLCMatches	M	RO	O	RO

Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevFilterIpDefault	M	RW	O	RW
docsDevFilterIpTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevFilterIpIndex	M	N-Acc	O	N-Acc
docsDevFilterIpStatus	M	RC	O	RC
docsDevFilterIpControl	M	RC	O	RC
docsDevFilterIpIfIndex	M	RC	O	RC
docsDevFilterIpDirection	M	RC	O	RC
docsDevFilterIpBroadcast	M	RC	O	RC
docsDevFilterIpSaddr	M	RC	O	RC
docsDevFilterIpSmask	M	RC	O	RC
docsDevFilterIpDaddr	M	RC	O	RC
docsDevFilterIpDmask	M	RC	O	RC
docsDevFilterIpProtocol	M	RC	O	RC
docsDevFilterIpSourcePortLow	M	RC	O	RC
docsDevFilterIpSourcePortHigh	M	RC	O	RC
docsDevFilterIpDestPortLow	M	RC	O	RC
docsDevFilterIpDestPortHigh	M	RC	O	RC
docsDevFilterIpMatches	M	RO	O	RO
docsDevFilterIpTos	M	RC	O	RC
docsDevFilterIpTosMask	M	RC	O	RC
docsDevFilterIpContinue	M	RC	O	RC
docsDevFilterIpPolicyId	M	RC	O	RC
docsDevFilterPolicyTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevFilterPolicyIndex	M	N-Acc	O	N-Acc
docsDevFilterPolicyId	M	RC	O	RC
docsDevFilterPolicyStatus	M	RC	O	RC
docsDevFilterPolicyPtr	M	RC	O	RC
docsDevFilterTosTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevFilterTosIndex	M	N-Acc	O	N-Acc
docsDevFilterTosStatus	M	RC	O	RC
docsDevFilterTosAndMask	M	RC	O	RC
docsDevFilterTosOrMask	M	RC	O	RC
docsDevCpeGroup				
NOTE: CM supporting IP spoofing function MUST implement this group. CM not supporting IP spoofing filter MUST NOT implement this group.				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevCpeEnroll	O	RW	N-Sup	
docsDevCpeIpMax	O	RW	N-Sup	

docsDevCpeTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
docsDevCpeIp	O	N-Acc	N-Sup	
docsDevCpeSource	O	RO	N-Sup	
docsDevCpeStatus	O	RC	N-Sup	
IP-MIB (RFC-2011)				
IP Group				
Objects	**CM**	**Access**	**CMTS**	**Access**
ipForwarding	M	RW	M	RW
ipDefaultTTL	M	RW	M	RW
ipInreceives	M	RO	M	RO
ipInHdrErrors	M	RO	M	RO
ipInAddrErrors	M	RO	M	RO
ipForwDatagrams	M	RO	M	RO
ipinUnknownProtos	M	RO	M	RO
ipInDiscards	M	RO	M	RO
ipInDelivers	M	RO	M	RO
ipOutRequests	M	RO	M	RO
ipOutDiscards	M	RO	M	RO
ipOutNoRoutes	M	RO	M	RO
ipReasmTimeout	M	RO	M	RO
ipReasmReqds	M	RO	M	RO
ipReasmOKs	M	RO	M	RO
ipReasmFails	M	RO	M	RO
ipFragOKs	M	RO	M	RO
ipFragFails	M	RO	M	RO
ipFragCreates	M	RO	M	RO
ipAddrTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
ipAdEntAddr	M	RO	M	RO
ipAdEntIfIndex	M	RO	M	RO
ipAdEntNetMask	M	RO	M	RO
ipAdEntBcastAddr	M	RO	M	RO
ipAdEntReasmMaxSize	M	RO	M	RO
IpNetToMediaTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
ipNetToMediaIfIndex	M	RC/RO	M	RC/RO
ipNetToMediaPhysAddress	M	RC/RO	M	RC/RO
ipNetToMediaNetAddress	M	RC/RO	M	RC/RO
ipNetToMediaType	M	RC/RO	M	RC/RO

Objects				
ipRoutingDiscards	M	RO	M	RO
ICMP Group				
Objects	**CM**	**Access**	**CMTS**	**Access**
icmpInMsgs	M	RO	M	RO
icmpInErrors	O	RO	M	RO
icmpInDestUnreachs	O	RO	M	RO
icmpInTimeExcds	O	RO	M	RO
icmpInParmProbs	O	RO	M	RO
icmpInSrcQuenchs	O	RO	M	RO
icmpInRedirects	O	RO	M	RO
icmpInEchos	M	RO	M	RO
icmpInEchosReps	O	RO	M	RO
icmpInTimestamps	O	RO	M	RO
icmpInTimeStampreps	O	RO	M	RO
icmpInAddrMasks	O	RO	M	RO
icmpInAddrMaskReps	O	RO	M	RO
icmpOutMsgs	M	RO	M	RO
icmpOutErrors	O	RO	M	RO
icmpOutDestUnreachs	O	RO	M	RO
icmpOutTimeExcds	O	RO	M	RO
icmpOutParmProbs	O	RO	M	RO
icmpOutSrcQuenchs	O	RO	M	RO
icmpOutRedirects	O	RO	M	RO
icmpOutEchos	O	RO	M	RO
icmpOutEchoReps	M	RO	M	RO
icmpOutTimestamps	O	RO	M	RO
icmpOutTimestampReps	O	RO	M	RO
icmpOutAddrMasks	O	RO	M	RO
icmpOutAddrMaskReps	O	RO	M	RO
UDP-MIB (RFC-2013)				
UDP Group				
Objects	**CM**	**Access**	**CMTS**	**Access**
udpInDatagrams	M	RO	M	RO
udpNoPorts	M	RO	M	RO
udpInErrors	M	RO	M	RO
udpOutDatagrams	M	RO	M	RO
UDP Listener Table				
Objects	**CM**	**Access**	**CMTS**	**Access**
udpLocalAddress	M	RO	M	RO
udpLocalPort	M	RO	M	RO

SNMPv2-MIB (RFC-3418)				
System Group				
Objects	**CM**	**Access**	**CMTS**	**Access**
sysDescr	M	RO	M	RO
sysObjectID	M	RO	M	RO
sysUpTime	M	RO	M	RO
sysContact	M	RW	M	RW
sysName	M	RW	M	RW
sysLocation	M	RW	M	RW
sysServices	M	RO	M	RO
sysORLastChange	M	RO	M	RO
sysORTable				
Object	**CM**	**Access**	**CMTS**	**Access**
sysORIndex	M	N-Acc	M	N-Acc
sysORID	M	RO	M	RO
sysORDescr	M	RO	M	RO
sysORUpTime	M	RO	M	RO
SNMP Group				
Objects	**CM**	**Access**	**CMTS**	**Access**
snmpInPkts	M	RO	M	RO
snmpInBadVersions	M	RO	M	RO
snmpOutPkts	Ob	RO	Ob	RO
snmpInBadCommunityNames	M	RO	M	RO
snmpInBadCommunityUses	M	RO	M	RO
snmpInASNParseErrs	M	RO	M	RO
snmpInTooBigs	Ob	RO	Ob	RO
snmpInNoSuchNames	Ob	RO	Ob	RO
snmpInBadValues	Ob	RO	Ob	RO
snmpInReadOnlys	Ob	RO	Ob	RO
snmpInGenErrs	Ob	RO	Ob	RO
snmpInTotalReqVars	Ob	RO	Ob	RO
snmpInTotalSetVars	Ob	RO	Ob	RO
snmpInGetRequests	Ob	RO	Ob	RO
snmpInGetNexts	Ob	RO	Ob	RO
snmpInSetRequests	Ob	RO	Ob	RO
snmpInGetResponses	Ob	RO	Ob	RO
snmpInTraps	Ob	RO	Ob	RO
snmpOutTooBigs	Ob	RO	Ob	RO
snmpOutNoSuchNames	Ob	RO	Ob	RO
snmpOutBadValues	Ob	RO	Ob	RO
snmpOutGenErrs	Ob	RO	Ob	RO
snmpOutGetRequests	Ob	RO	Ob	RO
snmpOutGetNexts	Ob	RO	Ob	RO

snmpOutSetRequests	Ob	RO	Ob	RO
snmpOutGetResponses	Ob	RO	Ob	RO
snmpOutTraps	Ob	RO	Ob	RO
snmpEnableAuthenTraps	M	RW	M	RW
snmpSilentDrops	M	RO	M	RO
snmpProxyDrops	M	RO	M	RO
Object	CM	Access	CMTS	Access
snmpSetSerialNo	M	RW	M	RW
Etherlike-MIB (RFC-2665)				
dot3StatsTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot3StatsIndex	M	RO	M	RO
dot3StatsAlignmentErrors	M	RO	M	RO
dot3StatsFCSErrors	M	RO	M	RO
dot3StatsSingleCollisionFrames	M	RO	M	RO
dot3StatsMultipleCollisionFrames	M	RO	M	RO
dot3StatsSQETestErrors	O	RO	O	RO
dot3StatsDeferredTransmissions	M	RO	M	RO
dot3StatsLateCollisions	M	RO	M	RO
dot3StatsExcessiveCollisions	M	RO	M	RO
dot3StatsInternalMacTransmitErrors	M	RO	M	RO
dot3StatsCarrierSenseErrors	O	RO	O	RO
dot3StatsFrameTooLongs	M	RO	M	RO
dot3StatsInternalMacReceiveErrors	M	RO	M	RO
dot3StatsEtherChipSet	D	RO	D	RO
dot3StatsSymbolErrors	M	RO	M	RO
dot3StatsDuplexStatus	M	RO	M	RO
dot3CollTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot3CollCount	O	NA	O	NA
dot3CollFrequencies	O	RO	O	RO
dot3ControlTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot3ControlFunctionsSupported	O	RO	O	RO
dot3ControlInUnknownOpcodes	O	RO	O	RO
dot3PauseTable				
Objects	**CM**	**Access**	**CMTS**	**Access**
dot3PauseAdminMode	O	RW	O	RW
dot3PauseOperMode	O	RO	O	RO
dot3InPauseFrames	O	RO	O	RO

dot3OutPauseFrames	O	RO	O	RO
USB-MIB				
NOTE: This MIB is required for CM that supports USB only.				
Object	**CM**	**Access**	**CMTS**	**Access**
usbNumber	M	RO	NA	
usbPortTable				
Object	**CM**	**Access**	**CMTS**	**Access**
usbPortIndex	M	RO	NA	
usbPortType	M	RO	NA	
usbPortRate	M	RO	NA	
usbDeviceTable				
Object	**CM**	**Access**	**CMTS**	**Access**
usbDeviceIndex	M	RO	NA	
usbDevicePower	M	RO	NA	
usbDeviceVendorID	M	RO	NA	
usbDeviceProductID	M	RO	NA	
usbDeviceNumberConfigurations	M	RO	NA	
usbDeviceActiveClass	M	RO	NA	
usbDeviceStatus	M	RO	NA	
usbDeviceEnumCounter	M	RO	NA	
usbDeviceRemoteWakeup	M	RO	NA	
usbDeviceRemoteWakeupOn	M	RO	NA	
usbCDCTable				
Object	**CM**	**Access**	**CMTS**	**Access**
usbCDCIndex	M	RO	NA	
usbCDCIfIndex	M	RO	NA	
usbCDCSubclass	M	RO	NA	
usbCDCVersion	M	RO	NA	
usbCDCDataTransferType	M	RO	NA	
usbCDCDataEndpoints	M	RO	NA	
usbCDCStalls	M	RO	NA	
usbCDCEtherTable				
Object	**CM**	**Access**	**CMTS**	**Access**
usbCDCEtherIndex	M	RO	NA	
usbCDCEtherIfIndex	M	RO	NA	
usbCDCEtherMacAddress	M	RO	NA	
usbCDCEtherPacketFilter	M	RO	NA	
usbCDCEtherDataStatisticsCapabilities	M	RO	NA	
usbCDCEtherDataCheckErrs	M	RO	NA	
DOCS-QOS-MIB (draft-ietf-ipcdn-qos-mib-04.txt)				
NOTE: 1.1 CM in 1.0 mode MUST NOT support this MIB.				

docsQosPktClassTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosPktClassId	M	N-Acc	M	N-Acc
docsQosPktClassDirection	M	RO	M	RO
docsQosPktClassPriority	M	RO	M	RO
docsQosPktClassIpTosLow	M	RO	M	RO
docsQosPktClassIpTosHigh	M	RO	M	RO
docsQosPktClassIpTosMask	M	RO	M	RO
docsQosPktClassIpProtocol	M	RO	M	RO
docsQosPktClassIpSourceAddr	M	RO	M	RO
docsQosPktClassIpSourceMask	M	RO	M	RO
docsQosPktClassIpDestAddr	M	RO	M	RO
docsQosPktClassIpDestMask	M	RO	M	RO
docsQosPktClassSourcePortStart	M	RO	M	RO
docsQosPktClassSourcePortEnd	M	RO	M	RO
docsQosPktClassDestPortStart	M	RO	M	RO
docsQosPktClassDestPortEnd	M	RO	M	RO
docsQosPktClassDestMacAddr	M	RO	M	RO
docsQosPktClassDestMacMask	M	RO	M	RO
docsQosPktClassSourceMacAddr	M	RO	M	RO
docsQosPktClassEnetProtocolType	M	RO	M	RO
docsQosPktClassEnetProtocol	M	RO	M	RO
docsQosPktClassUserPriLow	M	RO	M	RO
docsQosPktClassUserPriHigh	M	RO	M	RO
docsQosPktClassVlanId	M	RO	M	RO
docsQosPktClassState	M	RO	M	RO
docsQosPktClassPkts	M	RO	M	RO
docsQosPktClasBitMap	M	RO	M	RO
docsQosParamSetTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosParamSetServiceClassName	M	RO	M	RO
docsQosParamSetPriority	M	RO	M	RO
docsQosParamSetMaxTrafficRate	M	RO	M	RO
docsQosParamSetMaxTrafficBurst	M	RO	M	RO
docsQosParamSetMinReservedRate	M	RO	M	RO
docsQosParamSetMinReservedPkt	M	RO	M	RO
docsQosParamSetActiveTimeout	M	RO	M	RO
docsQosParamSetAdmittedTimeout	M	RO	M	RO
docsQosParamSetMaxConcatBurst	M	RO	M	RO
docsQosParamSetSchedulingType	M	RO	M	RO
docsQosParamSetNomPollInterval	M	RO	M	RO
docsQosParamSetTolPollJitter	M	RO	M	RO
docsQosParamSetUnsolicitGrantSize	M	RO	M	RO
docsQosParamSetNomGrantInterval	M	RO	M	RO
docsQosParamSetTolGrantJitter	M	RO	M	RO

docsQosParamSetGrantsPerInterval	M	RO	M	RO
docsQosParamSetTosAndMask	M	RO	M	RO
docsQosParamSetTosOrMask	M	RO	M	RO
docsQosParamSetMaxLatency	M	RO	M	RO
docsQosParamSetType	M	NA	M	NA
docsQosParamSetRequestPolicyOct	M	RO	M	RO
docsQosParamSetBitMap	M	RO	M	RO
docsQosServiceFlowTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosServiceFlowId	M	N-Acc	M	N-Acc
docsQosServiceFlowSID	M	RO	M	RO
docsQosServiceFlowDirection	M	RO	M	RO
docsQosServiceFlowPrimary	M	RO	M	RO
docsQosServiceFlowStatsTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosServiceFlowPkts	M	RO	M	RO
docsQosServiceFlowOctets	M	RO	M	RO
docsQosServiceFlowTimeCreated	M	RO	M	RO
docsQosServiceFlowTimeActive	M	RO	M	RO
docsQosServiceFlowPHSUnknowns	M	RO	M	RO
docsQosServiceFlowPolicedDropPkts	M	RO	M	RO
docsQosServiceFlowPolicedDelayPkts	M	RO	M	RO
docsQosUpstreamStatsTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosSID	N-Sup		M	N-Acc
docsQosUpstreamFragments	N-Sup		M	RO
docsQosUpstreamFragDiscards	N-Sup		M	RO
docsQosUpstreamConcatBursts	N-Sup		M	RO
docsQosDynamicServiceStatsTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosIfDirection	M	N-Acc	M	N-Acc
docsQosDSAReqs	M	RO	M	RO
docsQosDSARsps	M	RO	M	RO
docsQosDSAAcks	M	RO	M	RO
docsQosDSCReqs	M	RO	M	RO
docsQosDSCRsps	M	RO	M	RO
docsQosDSCAcks	M	RO	M	RO
docsQosDSDReqs	M	RO	M	RO
docsQosDSDRsps	M	RO	M	RO
docsQosDynamicAdds	M	RO	M	RO
docsQosDynamicAddFails	M	RO	M	RO

docsQosDynamicChanges	M	RO	M	RO
docsQosDynamicChangeFails	M	RO	M	RO
docsQosDynamicDeletes	M	RO	M	RO
docsQosDynamicDeleteFails	M	RO	M	RO
docsQosDCCReqs	M	RO	M	RO
docsQosDCCRsps	M	RO	M	RO
docsQosDCCAcks	M	RO	M	RO
docsQosDCCs	M	RO	M	RO
docsQosDCCFails	M	RO	M	RO
docsQosServiceFlowLogTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosServiceFlowLogIndex	N-Sup		M	N-Acc
docsQosServiceFlowLogIfIndex	N-Sup		M	RO
docsQosServiceFlowLogSFID	N-Sup		M	RO
docsQosServiceFlowLogCmMac	N-Sup		M	RO
docsQosServiceFlowLogPkts	N-Sup		M	RO
docsQosServiceFlowLogOctets	N-Sup		M	RO
docsQosServiceFlowLogTimeDeleted	N-Sup		M	RO
docsQosServiceFlowLogTimeCreated	N-Sup		M	RO
docsQosServiceFlowLogTimeActive	N-Sup		M	RO
docsQosServiceFlowLogDirection	N-Sup		M	RO
docsQosServiceFlowLogPrimary	N-Sup		M	RO
docsQosServiceFlowLogServiceClassName	N-Sup		M	RO
docsQosServiceFlowLogPolicedDropPkts	N-Sup		M	RO
docsQosServiceFlowLogPolicedDelayPkts	N-Sup		M	RO
docsQosServiceFlowLogControl	N-Sup		M	RW
docsQosServiceClassTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosServiceClassName	N-Sup		M	N-Acc
docsQosServiceClassStatus	N-Sup		M	RC
docsQosServiceClassPriority	N-Sup		M	RC
docsQosServiceClassMaxTrafficRate	N-Sup		M	RC
docsQosServiceClassMaxTrafficBurst	N-Sup		M	RC
docsQosServiceClassMinReservedRate	N-Sup		M	RC
docsQosServiceClassMinReservedPkt	N-Sup		M	RC
docsQosServiceClassMaxConcatBurst	N-Sup		M	RC
docsQosServiceClassNomPollInterval	N-Sup		M	RC
docsQosServiceClassTolPollJitter	N-Sup		M	RC
docsQosServiceClassUnsolicitGrantSize	N-Sup		M	RC
docsQosServiceClassNomGrantInterval	N-Sup		M	RC
docsQosServiceClassTolGrantJitter	N-Sup		M	RC
docsQosServiceClassGrantsPerInterval	N-Sup		M	RC
docsQosServiceClassMaxLatency	N-Sup		M	RC
docsQosServiceClassActiveTimeout	N-Sup		M	RC
docsQosServiceClassAdmittedTimeout	N-Sup		M	RC

docsQosServiceClassSchedulingTime	N-Sup		M	RC
docsQosServiceClassRequestPolicy	N-Sup		M	RC
docsQosServiceClassTosAndMask	N-Sup		M	RC
docsQosServiceClassTosOrMask	N-Sup		M	RC
docsQosServiceClassDirection	N-Sup		M	RC
docsQosServiceClassPolicyTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosServiceClassPolicyIndex	O	N-Acc	O	N-Acc
docsQosServiceClassPolicyName	O	RC	O	RC
docsQosServiceClassPolicyRulePriority	O	RC	O	RC
docsQosServiceClassPolicyStatus	O	RC	O	RC
docsQosPHSTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosPHSField	M	RO	O	RO
docsQosPHSMask	M	RO	O	RO
docsQosPHSSize	M	RO	O	RO
docsQosPHSVerify	M	RO	O	RO
docsQosPHSIndex	M	RO	O	RO
docsQosCmtsMacToSrvFlowTable				
Object	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsQosCmtsCmMac	N-Sup		M	N-Acc
docsQosCmtsServiceFlowId	N-Sup		M	N-Acc
docsQosCmtsIfIndex	N-Sup		M	RO
DOCS-SUBMGT-MIB (draft-ietf-ipcdn-subscriber-mib-02.txt) Subscriber Management MIB				
docsSubMgtCpeControlTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsSubMgtCpeControlMaxCpeIp	NA	NA	M	RW
docsSubMgtCpeControlActive	NA	NA	M	RW
docsSubMgtCpeControlLearnable	NA	NA	M	RW
docsSubMgtCpeControlReset	NA	NA	M	RW
docsSubMgtCpeMaxIpDefault	NA	NA	M	RW
docsSubMgtCpeActiveDefault	NA	NA	M	RW
docsSubMgtCpeIpTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsSubMgtCpeIpIndex	NA	NA	M	N-Acc
docsSubMgtCpeIpAddr	NA	NA	M	RO
docsSubMgtCpeIpLearned	NA	NA	M	RO

docsSubMgtPktFilterTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsSubMgtPktFilterGroup	NA	NA	M	N-Acc
docsSubMgtPktFilterIndex	NA	NA	M	N-Acc
docsSubMgtPktFilterSrcAddr	NA	NA	M	RC
docsSubMgtPktFilterSrcMask	NA	NA	M	RC
docsSubMgtPktFilterDstAddr	NA	NA	M	RC
docsSubMgtPktFilterDstMask	NA	NA	M	RC
docsSubMgtPktFilterUlp	NA	NA	M	RC
docsSubMgtPktFilterTosValue	NA	NA	M	RC
docsSubMgtPktFilterTosMask	NA	NA	M	RC
docsSubMgtPktFilterAction	NA	NA	M	RC
docsSubMgtPktFilterMatches	NA	NA	M	RO
docsSubMgtPktFilterStatus	NA	NA	M	RC
docsSubMgtTcpUdpFilterTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsSubMgtTcpUdpSrcPort	NA	NA	M	RC
docsSubMgtTcpUdpDstPort	NA	NA	M	RC
docsSubMgtTcpFlagValues	NA	NA	M	RC
docsSubMgtTcpFlagMask	NA	NA	M	RC
docsSubMgtTcpUdpStatus	NA	NA	M	RC
docsSubMgtCmFilterTable				
Object	**CM**	**Access**	**CMTS**	**Access**
docsSubMgtSubFilterDownstream	NA	NA	M	RW
docsSubMgtSubFilterUpstream	NA	NA	M	NW
docsSubMgtCmFilterDownstream	NA	NA	M	RW
docsSubMgtCmFilterUpstream	NA	NA	M	RW
Object	**CM**	**Access**	**CMTS**	**Access**
docsSubMgtSubFilterDownDefault	NA	NA	M	RW
docsSubMgtSubFilterUpDefault	NA	NA	M	RW
docsSubMgtCmFilterDownDefault	NA	NA	M	RW
docsSubMgtCmFilterUpDefault	NA	NA	M	RW
IGMP-STD-MIB (RFC-2933)				
This MIB is optional for Bridging CMTS				
NOTE: 1.1 CM in 1.0 mode is not required to implement RFC-2933				

IgmpInterfaceTable						
Object	**1.1 CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
IgmpInterfaceIfIndex	O	N-Acc	M	N-Acc	M	N-Acc
igmpInterfaceQueryInterval	O	RC	M	RC	M	RC
igmpInterfaceStatus	O	RC	M	RC	M	RC
igmpInterfaceVersion	O	RC	M	RC	M	RC
igmpInterfaceQuerier	O	RO	M	RO	M	RO

igmpInterfaceQueryMaxResponseTime	O	RO	M	RO	M	RO
igmpInterfaceVersion1QuerierTimer	O	RO	M	RO	M	RO
igmpInterfaceWrongVersionQueries	O	RO	M	RO	M	RO
igmpInterfaceJoins	O	RO	M	RO	M	RO
IgmpInterfaceGroups	O	RO	M	RO	M	RO
igmpInterfaceRobustness	O	RC	M	RC	M	RC
igmpInterfaceLastMembQueryIntvl	O	RC	M	RC	M	RC
igmpInterfaceProxyIfIndex	O	RC	M	RC	M	RC
igmpInterfaceQuerierUpTime	O	RO	M	RO	M	RO
igmpInterfaceQuerierExpiryTime	O	RO	M	RO	M	RO
igmpCacheTable						
Object	**1.1 CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
igmpCacheAddress	O	N-Acc	M	N-Acc	M	N-Acc
igmpCacheIfIndex	O	N-Acc	M	N-Acc	M	N-Acc
IgmpCacheSelf	O	RC	M	RC	M	RC
igmpCacheLastReporter	O	RO	M	RO	M	RO
igmpCacheUpTime	O	RO	M	RO	M	RO
igmpCacheExpiryTime	O	RO	M	RO	M	RO
igmpCacheStatus	O	RC	M	RC	M	RC
igmpCacheVersion1HostTimer	O	RO	M	RO	M	RO

DOCS-BPI-MIB RFC-3083						
docsBpiCmBaseTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiCmPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmPublicKey	M	RO	N-Sup		NA	
docsBpiCmAuthState	M	RO	N-Sup		NA	
docsBpiCmAuthKeySequenceNumber	M	RO	N-Sup		NA	
docsBpiCmAuthExpires	M	RO	N-Sup		NA	
docsBpiCmAuthReset	M	RW	N-Sup		NA	
docsBpiCmAuthGraceTime	M	RO	N-Sup		NA	
docsBpiCmTEKGraceTime	M	RO	N-Sup		NA	
docsBpiCmAuthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmReauthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmOpWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmRekeyWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRejectWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRequests	M	RO	N-Sup		NA	
docsBpiCmAuthReplies	M	RO	N-Sup		NA	
docsBpiCmAuthRejects	M	RO	N-Sup		NA	
docsBpiCmAuthInvalids	M	RO	N-Sup		NA	

docsBpiCmAuthRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorString	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorString	M	RO	N-Sup		NA	
docsBpiCmTEKTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiCmTEKPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmTEKState	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresOld	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresNew	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRequests	M	RO	N-Sup		NA	
docsBpiCmTEKKeyReplies	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejects	M	RO	N-Sup		NA	
docsBpiCmTEKInvalids	M	RO	N-Sup		NA	
docsBpiCmTEKAuthPends	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorStrin g	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorString	M	RO	N-Sup		NA	
docsBpiCmtsBaseTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiCmtsDefaultAuthLifetime	NA		NA		N-Sup	
docsBpiCmtsDefaultTEKLifetime	NA		NA		N-Sup	
docsBpiCmtsDefaultAuthGraceTime	NA		NA		N-Sup	
docsBpiCmtsDefaultTEKGraceTime	NA		NA		N-Sup	
docsBpiCmtsAuthRequests	NA		NA		N-Sup	
docsBpiCmtsAuthReplies	NA		NA		N-Sup	
docsBpiCmtsAuthRejects	NA		NA		N-Sup	
docsBpiCmtsAuthInvalids	NA		NA		N-Sup	
docsBpiCmtsAuthTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiCmtsAuthCmMacAddress	NA		NA		N-Sup	
docsBpiCmtsAuthCmPublicKey	NA		NA		N-Sup	
docsBpiCmtsAuthCmKeySequence Number	NA		NA		N-Sup	
docsBpiCmtsAuthCmExpires	NA		NA		N-Sup	
docsBpiCmtsAuthCmLifetime	NA		NA		N-Sup	
docsBpiCmtsAuthCmGraceTime	NA		NA		N-Sup	
docsBpiCmtsAuthCmReset	NA		NA		N-Sup	

docsBpiCmtsAuthCmRequests	NA		NA		N-Sup	
docsBpiCmtsAuthCmReplies	NA		NA		N-Sup	
docsBpiCmtsAuthCmRejects	NA		NA		N-Sup	
docsBpiCmtsAuthCmInvalids	NA		NA		N-Sup	
docsBpiCmtsAuthRejectErrorCode	NA		NA		N-Sup	
docsBpiCmtsAuthRejectErrorString	NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorCode	NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorString	NA		NA		N-Sup	
docsBpiCmtsTEKTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiCmtsTEKLifetime	NA		NA		N-Sup	
docsBpiCmtsTEKGraceTime	NA		NA		N-Sup	
docsBpiCmtsTEKExpiresOld	NA		NA		N-Sup	
docsBpiCmtsTEKExpiresNew	NA		NA		N-Sup	
docsBpiCmtsTEKReset	NA		NA		N-Sup	
docsBpiCmtsKeyRequests	NA		NA		N-Sup	
docsBpiCmtsKeyReplies	NA		NA		N-Sup	
docsBpiCmtsKeyRejects	NA		NA		N-Sup	
docsBpiCmtsTEKInvalids	NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorCode	NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorString	NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorCode	NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorString	NA		NA		N-Sup	
docsBpiIpMulticastMapTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiIpMulticastAddress	NA		NA		N-Sup	
docsBpiIpMulticastprefixLength	NA		NA		N-Sup	
docsBpiIpMulticastServiceId	NA		NA		N-Sup	
docsBpiIpMulticastMapControl	NA		NA		N-Sup	
docsBpiMulticastAuthTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpiMulticastServiceId	NA		NA		N-Sup	
docsBpiMulticastCmMacAddress	NA		NA		N-Sup	
docsBpiMulticastAuthControl	NA		NA		N-Sup	
DOCS-BPI2-MIB (draft-ietf-ipcdn-bpiplus-mib-05.txt)						
docsBpi2CmBaseTable						

Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmPrivacyEnable	O	RO	M	RO	NA	
docsBpi2CmPublicKey	O	RO	M	RO	NA	
docsBpi2CmAuthState	O	RO	M	RO	NA	
docsBpi2CmAuthKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresOld	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresNew	O	RO	M	RO	NA	
docsBpi2CmAuthReset	O	RW	M	RW	NA	
docsBpi2CmAuthGraceTime	O	RO	M	RO	NA	
docsBpi2CmTEKGraceTime	O	RO	M	RO	NA	
docsBpi2CmAuthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmReauthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmOpWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmRekeyWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmAuthRejectWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapMaxRetries	O	RO	M	RO	NA	
docsBpi2CmAuthentInfos	O	RO	M	RO	NA	
docsBpi2CmAuthRequests	O	RO	M	RO	NA	
docsBpi2CmAuthReplies	O	RO	M	RO	NA	
docsBpi2CmAuthRejects	O	RO	M	RO	NA	
docsBpi2CmAuthInvalids	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmTEKSAId	O	N-Acc	M	N-Acc	NA	
docsBpi2CmTEKSAType	O	RO	M	RO	NA	
docsBpi2CmTEKDataEncryptAlg	O	RO	M	RO	NA	
docsBpi2CmTEKDataAuthentAlg	O	RO	M	RO	NA	
docsBpi2CmTEKState	O	RO	M	RO	NA	
docsBpi2CmTEKKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresOld	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresNew	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRequests	O	RO	M	RO	NA	
docsBpi2CmTEKKeyReplies	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejects	O	RO	M	RO	NA	
docsBpi2CmTEKInvalids	O	RO	M	RO	NA	
docsBpi2CmTEKAuthPends	O	RO	M	RO	NA	

docsBpi2CmTEKKeyRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorString	O	RO	M	RO	NA	
docsBpi2CmIpMulticastMapTable						
Object	**1.1 CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmIpMulticastIndex	O	N-Acc	M	N-Acc	NA	
docsBpi2CmIpMulticastAddressType	O	RO	M	RO	NA	
docsBpi2CmIpMulticastAddress	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAId	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapState	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRequests	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapReplies	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejects	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmDeviceCertTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmDeviceCmCert	M	RW/RO	M	RW/RO	NA	
docsBpi2CmDeviceManufCert	M	RO	M	RO	NA	
docsBpi2CmCryptoSuiteTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmCryptoSuiteIndex	M	N-Acc	M	N-Acc	NA	
docsBpi2CmCryptoSuiteDataEncryptAlg	M	RO	M	RO	NA	
docsBpi2CmCryptoSuiteDataAuthentAlg	M	RO	M	RO	NA	
docsBpi2CmtsBaseEntryTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**

docsBpi2CmtsDefaultAuthLifetime	NA		NA		M	RW
docsBpi2CmtsDefaultTEKLifetime	NA		NA		M	RW
docsBpi2CmtsDefaultSelfSignedManufCertTrust	NA		NA		M	RW
docsBpi2CmtsCheckCertValidityPeriods	NA		NA		M	RW
docsBpi2CmtsAuthentInfos	NA		NA		M	RO
docsBpi2CmtsAuthRequests	NA		NA		M	RO
docsBpi2CmtsAuthReplies	NA		NA		M	RO
docsBpi2CmtsAuthRejects	NA		NA		M	RO
docsBpi2CmtsAuthInvalids	NA		NA		M	RO
docsBpi2CmtsSAMapRequests	NA		NA		M	RO
docsBpi2CmtsSAMapReplies	NA		NA		M	RO
docsBpi2CmtsSAMapRejects	NA		NA		M	RO
docsBpi2CmtsAuthEntryTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsAuthCmMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsAuthCmBpiVersion	NA		NA		M	RO
docsBpi2CmtsAuthCmPublicKey	NA		NA		M	RO
docsBpi2CmtsAuthCmKeySequenceNumber	NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresOld	NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresNew	NA		NA		M	RO
docsBpi2CmtsAuthCmLifetime	NA		NA		M	RW
docsBpi2CmtsAuthCmGraceTime	NA		NA		Ob	RO
docsBpi2CmtsAuthCmReset	NA		NA		M	RW
docsBpi2CmtsAuthCmInfos	NA		NA		M	RO
docsBpi2CmtsAuthCmRequests	NA		NA		M	RO
docsBpi2CmtsAuthCmReplies	NA		NA		M	RO
docsBpi2CmtsAuthCmRejects	NA		NA		M	RO
docsBpi2CmtsAuthCmInvalids	NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorString	NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorCode	NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorString	NA		NA		M	RO
docsBpi2CmtsAuthPrimarySAId	NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCertValid	NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCert	NA		NA		M	RO
docsBpi2CmtsTEKTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsTEKSAId	NA		NA		M	N-Acc
docsBpi2CmtsTEKSAType	NA		NA		M	RO

docsBpi2CmtsTEKDataEncryptAlg	NA		NA		M	RO
docsBpi2CmtsTEKDataAuthentAlg	NA		NA		M	RO
docsBpi2CmtsTEKLifetime	NA		NA		M	RW
docsBpi2CmtsTEKGraceTime	NA		NA		Ob	RO
docsBpi2CmtsTEKKeySequenceNumber	NA		NA		M	RO
docsBpi2CmtsTEKExpiresOld	NA		NA		M	RO
docsBpi2CmtsTEKExpiresNew	NA		NA		M	RO
docsBpi2CmtsTEKReset	NA		NA		M	RW
docsBpi2CmtsKeyRequests	NA		NA		M	RO
docsBpi2CmtsKeyReplies	NA		NA		M	RO
docsBpi2CmtsKeyRejects	NA		NA		M	RO
docsBpi2CmtsTEKInvalids	NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorString	NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorCode	NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorString	NA		NA		M	RO
docsBpi2CmtsIpMulticastMapTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsIpMulticastIndex	NA		NA		M	N-Acc
docsBpi2CmtsIpMulticastAddressType	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastAddress	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastMaskType	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastMask	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAId	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAType	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastDataEncryptAlg	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastDataAuthentAlg	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAMapRequests	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapReplies	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejects	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorString	NA		NA		M	RO
docsBpi2CmtsIpMulticastMapControl	NA		NA		M	RC/RO

docsBpi2CmtsMulticastAuthTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsMulticastAuthSAId	NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthCmMac Address	NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthControl	NA		NA		M	RC/RO
docsBpi2CmtsProvisionedCmCert Table						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsProvisionedCmCertM acAddress	NA		NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertTr ust	NA		NA		M	RC
docsBpi2CmtsProvisionedCmCertS ource	NA		NA		M	RO
docsBpi2CmtsProvisionedCmCertSt atus	NA		NA		M	RC
docsBpi2CmtsProvisionedCmCert	NA		NA		M	RC
docsBpi2CmtsCACertTable						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsCACertIndex	NA		NA		M	N-Acc
docsBpi2CmtsCACertSubject	NA		NA		M	RO
docsBpi2CmtsCACertIssuer	NA		NA		M	RO
docsBpi2CmtsCACertSerialNumber	NA		NA		M	RO
docsBpi2CmtsCACertTrust	NA		NA		M	RC
docsBpi2CmtsCACertSource	NA		NA		M	RO
docsBpi2CmtsCACertStatus	NA		NA		M	RC
docsBpi2CmtsCACert	NA		NA		M	RC
docsBpi2CmtsCACertThumprint	NA		NA		M	RO
docsBpi2CodeDownloadGroup						
Object	**1.1CM in 1.0 mode**	**Access**	**1.1 CM in 1.1 mode**	**Access**	**CMTS**	**Access**
docsBpi2CodeDownloadStatusCode	M	RO	M	RO	O	RO
docsBpi2CodeDownloadStatusStrin g	M	RO	M	RO	O	RO
docsBpi2CodeMfgOrgName	M	RO	M	RO	O	RO
docsBpi2CodeMfgCodeAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeMfgCvcAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerOrgName	M	RO	M	RO	O	RO

docsBpi2CodeCoSignerCodeAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerCvcAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeCvcUpdate	M	RW	M	RW	O	RW
SNMP-USM-DH-OBJECTS-MIB (RFC-2786)						
***NOTE:* SNMP-USM-DH-OBJECTS-MIB is only accessible when the device is in SNMP Coexistence Mode.**						
Object			**CM**	**Access**	**CMTS**	**Access**
usmDHParameters			M	RW	O	RW
usmDHUserKeyTable						
Object			**CM**	**Access**	**CMTS**	**Access**
usmDHUserAuthKeyChange			M	RC	O	RC
smDHUserOwnAuthKeyChange			M	RC	O	RC
usmDHUserPrivKeyChange			M	RC	O	RC
usmDHUserOwnPrivKeyChange			M	RC	O	RC
usmDHKickstartTable						
Object			**CM**	**Access**	**CMTS**	**Access**
usmDHKickstartIndex			M	N-Acc	O	N-Acc
usmDHKickstartMyPublic			M	RO	O	RO
usmDHKickstartMgrPublic			M	RO	O	RO
usmDHKickstartSecurityName			M	RO	O	RO
SNMP-VIEW-BASED-ACM-MIB (RFC-3415)						
(Note: SNMP-VIEW-BASED-ACM-MIB is ONLY accessible when the device is in SNMP Coexistence mode.)						
Object			**CM**	**Access**	**CMTS**	**Access**
vacmContextTable						
vacmContextName			M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
vacmSecurityToGroupTable						
vacmSecurityModel			M	N-Acc	M	N-Acc
vacmSecurityName			M	N-Acc	M	N-Acc
vacmGroupName			M	RC	M	RC
vacmSecurityToGroupStorageType			M	RC	M	RC
vacmSecurityToGroupStatus			M	RC	M	RC

Object	CM	Access	CMTS	Access
vacmAccessTable				
vacmAccessContextPrefix	M	N-Acc	M	N-Acc
vacmAccessSecurityModel	M	N-Acc	M	N-Acc
vacmAccessSecurityLevel	M	N-Acc	M	N-Acc
vacmAccessContextMatch	M	RC	M	RC
vacmAccessReadViewName	M	RC	M	RC
vacmAccessWriteViewName	M	RC	M	RC
vacmAccessNotifyViewName	M	RC	M	RC
vacmAccessStorageType	M	RC	M	RC
vacmAccessStatus	M	RC	M	RC
vacmViewSpinLock	M	RW	M	RW
Object	**CM**	**Access**	**CMTS**	**Access**
vacmViewTreeFamilyTable				
vacmViewTreeFamilyViewName	M	N-Acc	M	N-Acc
vacmViewTreeFamilySubtree	M	N-Acc	M	N-Acc
vacmViewTreeFamilyMask	M	RC	M	RC
vacmViewTreeFamilyType	M	RC	M	RC
vacmViewTreeFamilyStorageType	M	RC	M	RC
vacmViewTreeFamilyStatus	M	RC	M	RC
SNMP-COMMUNITY-MIB (RFC-2576)				
(Note: SNMP-COMMUNITY-MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpCommunityTable				
snmpCommunityIndex	M	N-Acc	M	N-Acc
snmpCommunityName	M	RC	M	RC
snmpCommunitySecurityName	M	RC	M	RC
snmpCommunityContextEngineID	M	RC	M	RC
snmpCommunityContextName	M	RC	M	RC
snmpCommunityTransportTag	M	RC	M	RC
snmpCommunityStorageType	M	RC	M	RC
snmpCommunityStatus	M	RC	M	RC
Object	**CM**	**Access**	**CMTS**	**Access**
SnmpTargetExtTable				
snmpTargetAddrTMask	M	RC	M	RC
snmpTargetAddrMMS	M	RC	M	RC

snmpTrapAddress	O	ACC-FN	O	ACC-FN
snmpTrapCommunity	O	ACC-FN	O	ACC-FN
SNMP Management Framework architecture (RFC-3411)				
(Note: SNMP Management Framework architecture MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpEngine Group				
snmpEngineID	M	RO	M	RO
snmpEngineBoots	M	RO	M	RO
snmpEngineTime	M	RO	M	RO
snmpEngineMaxMessageSize	M	RO	M	RO
SNMP Message Processing and Dispatching MIB (RFC-3412)				
(Note: SNMP Message Processing and Dispatching MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpMPDStats				
snmpUnknownSecurityModels	M	RO	M	RO
snmpInvalidMsgs	M	RO	M	RO
snmpUnknownPDUHandlers	M	RO	M	RO
(RFC-3413)				
(Note: RFC-3413 is ONLY accessible when the device is in SNMP Coexistence mode.)				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpTargetSpinLock	M	RW	M	RW
snmpTargetAddrTable				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpTargetAddrName	M	N-Acc	M	N-Acc
snmpTargetAddrTDomain	M	RC	M	RC
SnmpTargetAddrTAddress	M	RC	M	RC
snmpTargetAddrTimeout	M	RC	M	RC
snmpTargetAddrRetryCount	M	RC	M	RC
snmpTargetAddrTagList	M	RC	M	RC
snmpTargetAddrParams	M	RC	M	RC

snmpTargetAddrStorageType	M	RC	M	RC
snmpTargetAddrRowStatus	M	RC	M	RC
snmpTargetParamsTable				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpTargetParamsName	M	N-Acc	M	N-Acc
snmpTargetParamsMPModel	M	RC	M	RC
snmpTargetParamsSecurityModel	M	RC	M	RC
snmpTargetParamsSecurityName	M	RC	M	RC
snmpTargetParamsSecurityLevel	M	RC	M	RC
snmpTargetParamsStorageType	M	RC	M	RC
snmpTargetParamsRowStatus	M	RC	M	RC
snmpUnavailableContexts		RO	M	RO
snmpUnknownContexts	M	RO	M	RO
snmpNotifyTable				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpNotifyName	M	N-Acc	M	N-Acc
snmpNotifyTag	M	RC	M	RC
snmpNotifyType	M	RC	M	RC
snmpNotifyStorageType	M	RC	M	RC
SnmpNotifyRowStatus	M	RC	M	RC
snmpNotifyFilterProfileTable				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpNotifyFilterProfileName	M	RC	M	RC
snmpNotifyFilterProfileStorType	M	RC	M	RC
snmpNotifyFilterProfileRowStatus	M	RC	M	RC
snmpNotifyFilterTable				
Object	**CM**	**Access**	**CMTS**	**Access**
snmpNotifyFilterSubtree	M	N-Acc	M	N-Acc
snmpNotifyFilterMask	M	RC	M	RC
snmpNotifyFilterType	M	RC	M	RC
snmpNotifyFilterStorageType	M	RC	M	RC
snmpNotifyFilterRowStatus	M	RC	M	RC
SNMP-USER-BASED-SM-MIB (RFC-3414)				
(Note: RFC-3414 MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
usmStats				

Object			**CM**	**Access**	**CMTS**	**Access**
usmStatsUnsupportedSecLevels			M	RO	M	RO
usmStatsNotInTimeWindows			M	RO	M	RO
usmStatsUnknownUserNames			M	RO	M	RO
usmStatsUnknownEngineIDs			M	RO	M	RO
usmStatsWrongDigests			M	RO	M	RO
usmStatsDecryptionErrors			M	RO	M	RO
usmUser						
Object			**CM**	**Access**	**CMTS**	**Access**
usmUserSpinLock			M	RW	M	RW
usmUserTable						
Object			**CM**	**Access**	**CMTS**	**Access**
usmUserEngineID			M	N-Acc	M	N-Acc
usmUserName			M	N-Acc	M	N-Acc
usmUserSecurityName			M	RO	M	RO
usmUserCloneFrom			M	RC	M	RC
usmUserAuthProtocol			M	RC	M	RC
usmUserAuthKeyChange			M	RC	M	RC
usmUserOwnAuthKeyChange			M	RC	M	RC
usmUserPrivProtocol			M	RC	M	RC
usmUserPrivKeyChange			M	RC	M	RC
usmUserOwnPrivKeyChange			M	RC	M	RC
usmUserPublic			M	RC	M	RC
usmUserStorageType			M	RC	M	RC
usmUserStatus			M	RC	M	RC
DOCS-IF-EXT-MIB	**1.1CM in 1.0 Mode**	**Access**	**1.1 CM in 1.1 Mode**	**Access**	**CMTS**	**Access**
docsIfDocsisCapability	D	RO	D	RO	D	RO
docsIfDocsisOperMode	D	RO	D	RO	D	RO
docsIfCmtsCmStatusDocsisMode	N/A		N/A		D	NA
DOCS-CABLE-DEVICE-TRAP-MIB	**1.1CM in 1.0 Mode**	**Access**	**1.1 CM in 1.1 Mode**	**Access**	**CMTS**	**Access**
docsDevCmTrapControl	O	RW	M	RW	NA	
docsDevCmtsTrapControl	NA		NA		M	RW
docsDevCmInitTLVUnknownTrap	NA		M	ATRAP	NA	
docsDevCmDynServReqFailTrap	NA		M	ATRAP	NA	
docsDevCmDynServRspFailTrap	NA		M	ATRAP	NA	
docsDevCmDynServAckFailTrap	NA		M	ATRAP	NA	

docsDevCmBpiInitTrap	NA		M	ATRAP	NA	
docsDevCmBPKMTrap	NA		M	ATRAP	NA	
docsDevCmDynamicSATrap	NA		M	ATRAP	NA	
docsDevCmDHCPFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeInitTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeSuccessTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeCVCFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmTODFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmDCCReqFailTrap	O	ATRAP	M	ATRAP		
docsDevCmDCCRspFailTrap	O	ATRAP	M	ATRAP		
docsDevCmDCCAckFailTrap	O	ATRAP	M	ATRAP		
docsDevCmtsInitRegReqFailTrap			NA		M	ATRAP
docsDevCmtsInitRegRspFailTrap			NA		M	ATRAP
docsDevCmtsInitRegAckFailTrap			NA		M	ATRAP
docsDevCmtsDynServReqFailTrap			NA		M	ATRAP
docsDevCmtsDynServRspFailTrap			NA		M	ATRAP
docsDevCmtsDynServAckFailTrap			NA		M	ATRAP
docsDevCmtsBpiInitTrap			NA		M	ATRAP
docsDevCmtsBPKMTrap			NA		M	ATRAP
docsDevCmtsDynamicSATrap			NA		M	ATRAP
docsDevCmtsDCCReqFailTrap			NA		M	ATRAP
docsDevCmtsDCCRspFailTrap			NA		M	ATRAP
docsDevCmtsDCCAckFailTrap			NA		M	ATRAP

Appendix B. RFC-2863 ifTable MIB-Object details

Table 25. RFC-2863 ifTable MIB-Object details

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifIndex:"A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. [The Primary CPE MUST be Interface number 1] The value for each interface sub-layer must remain constant at least from one reinitialization of the entity's network management system to the next reinitialization."	(n)	(n)	(n)	(n)	[1 or 4+(n)]	2	3	4	[1 or 4+(n)]	[1 or 4+(n)]
ifType:"The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention."	6	127	128	129	6	127	128	129	160	(IANA num)
ifSpeed:"An estimate of the interface's current bandwidth in bits per second. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object should be zero."	10,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	10,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	12,000,000	speed

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifHighSpeed:"An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n' then the speed of the interface is somewhere in the range of `n-500,000' to `n+499,999'. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object should be zero."	10	0	~64-QAM=30, ~256-QUAM=42	(n)	10	0	~64-QAM=30, ~256-QUAM=42	(n)	12	speed
ifPhysAddress:"The interface's address at its protocol sub-layer. [For RF Upstream/Downstream; return empty string. For MAC Layer; return the physical address of this interface.] For example, for an 802.x interface, this object normally contains a MAC address. The interface's media-specific MIB must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length."	Enet-MAC	CATV-MAC	Empty-String	Empty-String	Enet-MAC	CATV-MAC	Empty-String	Empty-String	USB-Phys Addr.	Phys Addr.

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifAdminStatus:"The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the up(1) state. As a result of either explicit management action, ifAdminStatus is then changed to either the down(2) or testing(3) states (or remains in the up(1) state). [For CM: When a managed system initializes, all interfaces start with ifAdminStatus in the up(1) state. As a result of explicit management action, ifAdminStatus is then changed to either the down(2) or testing(3) states (or remains in the up(1) state). For CMTS: When a managed system initializes, all interface start with ifAdminStatus in the up(1) state. As a result of either explicit management or configuration information the saved via other non SNMP method (i.e. CLI commands) retained by the managed system, ifAdminStatus is then changed to either the down(2) or testing(3) states (or remains in the up(1) state).]"	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)
ifOperStatus:"The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down(2) then ifOperStatus should be down(2). If ifAdminStatus is changed to up(1) then ifOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(6) state if the interface has missing (typically, hardware) components."	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifMtu:"The size of the largest packet which can be sent/received on the interface, specified in octets. [For RF Upstream/Downstream; the value includes the length of the MAC header. For MAC Layer; return 1500.] For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface."	1500	1500	1764	1764	1500	1500	1764	1764	1500	1500?
ifInOctets:"The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers. For MAC; The total number of data octets (bridge data, data target for the managed device) received on this interface from RF-downstream interface and before application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n)	(n)

* The ifEntry for Downstream interfaces supports the ifGeneralInformationGroup and the ifPacketGroup of the Interfaces MIB. This is an output only interface at the CMTS and all input status counters – ifIn* - will return zero. This is an input only interface at the CM and all output status counters – ifOut* - will return zero. The ifEntry for Upstream interfaces supports the ifGeneralInformationGroup and the ifPacketGroup of the Interfaces MIB. This is an input only interface at the CMTS and all output status counters – ifOut* - will return zero. This is an output only interface at the CM and all input status counters – ifIn* - will return zero.

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
IfHCInOctets: (usage**) "The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers.] This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count[76]	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutOctets:"The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface. For MAC; The total number of data octets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

** For interfaces that operate at 20,000,000 (20 million) bits per second or less, 32-bit byte and packet counters MUST be used. For interfaces that operate faster than 20,000,000 bits/second, and slower than 650,000,000 bits/second, 32-bit packet counters MUST be used and 64-bit octet counters MUST be used. For interfaces that operate at 650,000,000 bits/second or faster, 64-bit packet counters AND 64-bit octet counters MUST be used. When 64-bit counters are in use, the 32-bit counters MUST still be available. The 32-bit counters report the low 32-bits of the associated 64-bit count (e.g., ifInOctets will report the least significant 32 bits of ifHCInOctets). This enhances inter-operability with existing implementations at a very minimal cost to agents.

*** If the optional 64-bit counter is implemented then the corresponding 32-bit counter MUST represent the low 32-bits of the associated 64-bit counter.

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifHCOutOctets: (usage**) "The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface.] This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifInUcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Unicast data packets (bridge data, data target for the managed device) received on this interface from RF-downstream interface before application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifHCInUcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Unicast data packets (bridge data, data target for the managed device) received on this interface from RF-downstream interface before application of protocol filters defined in RFC-2669.] This object is a 64-bit version of ifInUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifInMulticastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Multicast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifHCInMulticastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Multicast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifInBroadcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets received on this interface, targeted for upper protocol layers. For MAC layer; The number of Broadcast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifHCInBroadcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets received on this interface, targeted for upper protocol layers. For MAC layer; The number of Broadcast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] This object is a 64-bit version of ifInBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifInDiscards:"The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
ifInErrors:"For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifInUnknownProtos:"For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter will always be 0. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
ifOutUcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Unicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifHCOutUcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Unicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] This object is a 64-bit version of ifOutUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutMulticastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Multicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifHCOutMulticastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Multicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifOutMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutBroadcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Broadcast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifHCOutBroadcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Broadcast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] This object is a 64-bit version of ifOutBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutDiscards:"The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)
ifOutErrors:"For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 10 Meg Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifPromiscuousMode:"This object has a value of false(2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true(1) when the station accepts all packets/frames transmitted on the media. The value true(1) is only legal on certain types of media. If legal, setting this object to a value of true(1) may require the interface to be reset before becoming effective. The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface."	true(1) false(2)	true(1) false(2)	false(2)	true(1) false(2)	true(1) false(2)	true(1) false(2)	true(1) false(2)	false(2)	true(1) false(2)	true(1) false(2)

RFC-2863 MIB-Object details for Cable Device using 100 Meg Ethernet (effected MIB-Objects only; all others same as above table)	CMTS-Ethernet-100	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-100	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifSpeed:"An estimate of the interface's current bandwidth in bits per second. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object should be zero."	100,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	100,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	12,000,000	speed

RFC-2863 MIB-Object details for Cable Device using 100 Meg Ethernet **(effected MIB-Objects only; all others same as above table)**	**CMTS-Ethernet-100**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-100**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifHighSpeed:"An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n' then the speed of the interface is somewhere in the range of `n-500,000' to `n+499,999'. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object should be zero."	100	0	~64-QAM=30, ~256-QUAM=42	(n)	100	0	~64-QAM=30, ~256-QUAM=42	(n)	12	speed
ifInOctets:"The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers. For MAC; The total number of data octets (bridge data, data target for the managed device) received on this interface from RF-downstream interface and before application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = low 32-bits of the 64-bit count	(n)	MUST be 0	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 100 Meg Ethernet **(effected MIB-Objects only; all others same as above table)**	**CMTS-Ethernet-100**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-100**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
IfHCInOctets: (usage**) "The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers.] This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = 64-bit count	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	(n) = 64-bit count	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutOctets:"The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface. For MAC; The total number of data octets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n) = low 32-bits of the 64-bit count	(n)	MUST be 0	(n)	(n)	(n)

RFC-2863 MIB-Object details for Cable Device using 100 Meg Ethernet **(effected MIB-Objects only; all others same as above table)**	**CMTS-Ethernet-100**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream**	**CM-Ethernet-100**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPE Other Type**
ifHCOutOctets: (usage**) "The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface.] This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = 64-bit count	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	(n) = 64-bit count	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***

Appendix C. RFC-1493 and RFC-2863 MIB-Object Details for CCCM[77]

For MIB objects in RFC-1493 and RFC-2863 to be tested in applicable ATPs, they MUST be interpreted according this appendix.

C.1 RFC-1493 MIB-Object Details

Table 26. RFC-1493 MIB-Object Details

BRIDGE-MIB (RFC-1493)		
dot1dBase group		
Objects	**CCCM**	**Access**
dot1dBaseBridgeAddress	M	RO
dot1dBaseNumPorts	M	RO
dot1dBaseType	M	RO
dot1dBasePortTable		
Objects	**CCCM**	**Access**
dot1dBasePort	M	RO
dot1dBasePortIfIndex	M	RO
dot1dBasePortCircuit	M	RO
dot1dBasePortDelayExceededDiscards	M	RO
dot1dBasePortMtuExceededDiscards	M	RO
dot1dStp group		
Objects	**CCCM**	**Access**
dot1dStpProtocolSpecification	NA	
dot1dStpPriority	NA	
dot1dStpTimeSinceTopologyChange	NA	
dot1dStpTopChanges	NA	
dot1dStpDesignatedRoot	NA	
dot1dStpRootCost	NA	
dot1dStpRootPort	NA	
dot1dStpMaxAge	NA	
dot1dStpHelloTime	NA	
dot1dStpHoldTime	NA	
dot1dStpForwardDelay	NA	
dot1dStpBridgeMaxAge	NA	
dot1dStpBridgeHelloTime	NA	

[77] Appendix C changes per ECN OSS-N-02190 by GO on 11/15/02.

dot1dStpBridgeForwardDelay	NA	
dot1dStpPortTable	NA	
Objects	**CCCM**	**Access**
dot1dStpPort	NA	
dot1dStpPortPriority	NA	
dot1dStpPortState	NA	
dot1dStpPortEnable	NA	
dot1dStpPortPathCost	NA	
dot1dStpPortDesignatedRoot	NA	
dot1dStpPortDesignatedCost	NA	
dot1dStpPortDesignatedBridge	NA	
dot1dStpPortDesignatedPort	NA	
dot1dStpPortForwardTransitions	NA	
dot1dTp group		
Objects	**CCCM**	**Access**
dot1dTpLearnedEntryDiscards	M	RO
dot1dTpAgingTime	M	RO
dot1dTpFdbTable		
Objects	**CCCM**	**Access**
dot1dTpFdbAddress	M	RO
dot1dTpFdbPort	M	RO
dot1dTpFdbStatus	M	RO
dot1dTpPortTable		
Objects	**CCCM**	**Access**
dot1dTpPort		
dot1dTpPortMaxInfo	M	RO
dot1dTpPortInFrames	M	RO
dot1dTpPortOutFrames	M	RO
dot1dTpPortInDiscards	M	RO
dot1dStaticTable		
Objects	**CCCM**	**Access**
dot1dStaticAddress	O	RO
dot1dStaticReceivePort	O	RO
dot1dStaticAllowedToGoTo	O	RO
dot1dStaticStatus	O	RO

C.2 Implementation of RFC-1493 MIB for CCCM

The dot1dBase Group

This is a mandatory group which contains the objects which are applicable to all types of bridges.

Table 27. The dot1dBase Group.

Mib Object	Object Valve Description	Access
dot1dBaseBridgeAddress	CCCM MAC address	Hardcoded, read-only
dot1dBaseNumPorts	2 (RF port, CPE port)	Hardcoded, read-only
dot1dBaseType	Transparent-only(2)	Hardcoded, read-only
dot1dBasePortTable	See the Table Below	

Dot1dBasePortTable

The following table contains generic information about every port that is associated**.**

Table 28. Dot1dBasePortTable.

Mib Object	Object Valve Description	Access
Dot1dBasePort	1 – for CPE port; 2 – for RF port	Read-only
Dot1dBasePortIfIndex	IfIndex of CPE interface (1) – for CPE port; IfIndex of CATV MAC interface (2) – for RF port	Read-only
Dot1dBasePortCircuit	{0,0} – For a port which has a unique value of dot1dBasePortIfIndex, this object can have the value{0,0} .	Read-only
Dot1dBasePortDelayExceededDiscards	# of frames discarded by the port due to excessive transit delay through the bridge, May be 0 .	Read-only
Dot1dBasePortMtuExceededDiscards	# of frames discarded by the port due to excessive size, May be 0.	Read-only

The dot1dStp Group = Not Implemented

If a node does not implemented the Spanning Tree Protocol, this group will not be implemented.

The dot1dSr Group = Not Implemented

If source routing is not supported this group will not be implemented.

The dot1dTp Group = Not implemented or limited implementation as described below:

This group contains objects that describe the entity's state with respect to transparent bridging. If transparent bridging is not supported this group will not be implemented. This group is applicable to transparent only and SRT bridges.

Table 29. The dot1dTp Group.

Mib Object	Object Valve Description	Access
dot1dTpLearnedEntryDiscards	Supported	Hardcoded, readonly
dot1dTpAgingTime	1000001	Hardcoded, readonly
dot1dTpFdbEntry (Table)	For transparent bridging only – read-only Table has 2 entries, see below	
dot1dTpFdbAddress	CPE MAC address .	Hardcoded, readonly
dot1dTpFdbPort	0 (port number has not been learned)	Hardcoded, readonly
dot1dTpFdbStatus	4: self(4)	Hardcoded, readonly
dot1dTpPortTable	See Table Below	

Table 30. dot1dFdbTable.

Mib Object	Object Valve Description	Access
dot1dTpFdbAddress	CPE MAC address – for port on CATV MAC interface; CATV MAC address – for port on CPE interface;	Hard coded, read-only
dot1dTpFdbPort	0 (port number has not been learned) for both entries	Hard coded, read-only
dot1dTpFdbStatus	self(4) for both entries	Hard coded, read-only

dot1dTpPortTable

A table that contains information about every port that is associated with the transparent bridge.

Table 31. dot1dTpPortTable.

Mib Object	Object Valve Description	Access
Dot1dTpPortMaxInfo	1500 – Maximum size of the info(non-mac) field that the port will receive or transmit.	read-only
Dot1dTpPortInFrame	Counter - supported	read-only
Dot1dTpPortOutFrames	Counter – supported	read-only
Dot1dTpPortInDiscards	CPE = CPE Discards MAC=MAC Discards	read-only

The dot1dStatic Group = Not Implemented, implementation of this group is optional.

C.2.1 RFC-2863 ifTable MIB-Object details for CCCM

From SNMP perspective, CCCM MUST mimics the standalone CM. The generic network interface MIB on logical CPE interface MUST be supported (RFC-2863), with the following recommended values

Table 32. RFC-2863 ifTable MIB-Object details for CCCM.

ifTable / ifXTable Table Column	Implementation for CPE interface
ifIndex	1
ifDescr	Up to RFC-2863 and OSSI Appendix B
ifType	1 - other
ifMtu	1500
ifSpeed	10Mbit/sec
ifPhysAddress	Empty string
ifAdminStatus	Up to RFC-2863. Setting this object to ‘disable’ causes no data flow to the PC CPE behind the modem. (Similar to the “NACO off” operation)
ifOperStatus	Up to RFC-2863 and OSSI Appendix B
ifLastChange	Up to RFC-2863 and OSSI Appendix B
ifInOctets	Up to RFC-2863 and OSSI Appendix B
ifInUcastPkts	Up to RFC-2863 and OSSI Appendix B
ifInDiscards	Up to RFC-2863 and OSSI Appendix B
ifInErrors	Up to RFC-2863 and OSSI Appendix B
ifInUnknownProtos	Up to RFC-2863 and OSSI Appendix B
ifOutOctets	Up to RFC-2863 and OSSI Appendix B
ifOutUcastPkts	Up to RFC-2863 and OSSI Appendix B
ifOutDiscards	Up to RFC-2863 and OSSI Appendix B
ifOutErrors	Up to RFC-2863 and OSSI Appendix B
ifName	“Textual description”
ifInMulticastPkts	Up to RFC-2863 and OSSI Appendix B
ifInBroadcastPkts	Up to RFC-2863 and OSSI Appendix B
ifOutMulticastPkts	Up to RFC-2863 and OSSI Appendix B
ifOutBroadcastPkts	Up to RFC-2863 and OSSI Appendix B
ifHCInOctets	Up to RFC-2863 and OSSI Appendix B
ifHCInUcastPkts	Up to RFC-2863 and OSSI Appendix B
ifHCInMulticastPkts	Up to RFC-2863 and OSSI Appendix B
ifHCInBroadcastPkts	Up to RFC-2863 and OSSI Appendix B
ifHCOutOctets	Up to RFC-2863 and OSSI Appendix B
ifHCOutUcastPkts	Up to RFC-2863 and OSSI Appendix B
ifHCOutMulticastPkts	Up to RFC-2863 and OSSI Appendix B
ifHCOutBroadcastPkts	Up to RFC-2863 and OSSI Appendix B
ifLinkUpDownTrapEnable	Up to RFC-2863 and OSSI Appendix B, enabled by default
ifHighSpeed	10Mbit/sec
ifPromiscuousMode	TRUE, read-only access
ifConnectorPresent	Always True(1)
ifAlias	Up to RFC-2863 and OSSI Appendix B
ifCounterDiscontinuityTime	Up to RFC-2863 and OSSI Appendix B

Appendix D. Business Process Scenarios For Subscriber Account Management

In order to develop the DOCS-OSS Subscriber Account Management Specification, it is necessary to consider high-level business processes common to cable operators and the associated operational scenarios. The following definitions represent a generic view of key processes involved. It is understood that business process terminology varies among different cable operators, distinguished by unique operating environments and target market segments

For the purpose of this document, Subscriber Account Management refers to the following business processes and terms:

Class of Service Provisioning Processes, which are involved in the automatic and dynamic provisioning and enforcement of subscribed class of policy-based service level agreements (SLAs);

Usage-Based Billing Processes, which are involved in the processing of bills based on services rendered to and consumed by paying subscriber customers.

D.1 The Old Service Model: "One Class Only" & "Best Effort" Service

The Internet is an egalitarian cyber society in its pure technical form where all Internet Protocol (IP) packets are treated as equals. Given all IP packets have equal right of way over the Internet, it is a "one class fits all", "first come, first serve" type of service level arrangement. The response time and quality of delivery service is promised to be on a "best effort" basis only.

Unfortunately, while all IP packets are theoretically equal, certain classes of IP packets must be processed differently. When transmitting data packets, traffic congestion causes no fatal problems except unpredictable delays and frustrations. However, in a convergent IP world where data packets are mixed with those associated with voice and streaming video, such "one class" service level and "best effort only" quality is not workable.

D.2 The Old Billing Model: "Flat Rate" Access

As high speed data over cable service deployment moves to the next stage, serious considerations must be made by all cable operators to abandon old business practices, most notably "flat rate" fee structure. No service provider can hope to stay in business long by continuing to offer a single, "flat rate" access service to all subscribers, regardless of actual usage.

Imagine your utility bills were the same month after month, whether you used very little water or electricity every day, or if you ran your water and your air conditioning at full blast 24 hours a day. You are entitled, just like everyone else, to consume as much or as little as you wished, anytime you wanted it. Chances are you would not accept such a service agreement. Not only because it is not a fair arrangement, but also because such wasteful consumption would put pressure on the finite supply of water and electricity that most of your normal demands for usage would likely go unfulfilled.

D.3 A Successful New Business Paradigm

The new paradigm for delivering IP-based services over cable networks is forcing all cable operators to adopt a new business paradigm. The retention of customers will require that an operator offer different class of service options and associated access rates with guaranteed provisioning and delivery of subscribed services. "Back Office" usage-based accounting and subscriber billing will become an important competitive differentiation in the emergence of high-speed data over cable services.

D.3.1 Integrating "Front End" Processes Seamlessly with "Back Office" Functions

A long-standing business axiom states that accountability exists only with the right measurements and that business prospers only with the proper management information. An effective subscriber account management system for data over cable services should meet three (3) major requirements:

Automatic & Dynamic Subscriber Provisioning

The 1st requirement is to integrate service subscription orders and changes automatically and dynamically, with the various processes that invoke the provisioning and delivering of subscribed and/or "on demand" services;

Guaranteed Class & Quality of Services

The 2nd requirement is to offer different class of services with varying rates and guarantee the quality of service level associated with each service class;

Data Collection, Warehousing & Usage Billing

The 3rd requirement is to capture a subscriber's actual usage, calculating the bill based on the rate associated with the customer's subscribed service levels.

D.3.2 Designing Class of Services

While designing different class of service offerings, a cable operator might consider the following framework:

Class of Service by Account Type – Business vs. Residential Accounts

Class of Service by Guaranteed Service Levels

Class of Service by Time of Day and/or Day of Week

"On Demand" Service by Special Order

The following is a plausible sample of class of services:

- *"Best Effort" Service Without Minimum Guarantee*

  This class of "Best Effort Only" service is the normal practice of today where subscribers of this class of service are allocated only excess channel bandwidth available at the time while each subscriber's access is capped at a maximum bandwidth (for example at 512 kilobit per second).

- *Platinum Service for Business and High-Access Residential Accounts*

  Business accounts subscribing to this service are guaranteed a minimum data rate of downstream bandwidth – 512 kilobit per second – and if excess bandwidth is available, they are allowed to burst to 10 megabit per second.

- *Gold Service for Business Accounts*

  This class of service guarantees subscribers a 256 kilobit per second downstream data rate during business hours (for example from 8 a.m. to 6 p.m.) and 128 kilobit per second at other times. If excess bandwidth is available at any time, data is allowed to burst to 5 megabit per second.

- *Gold Service for Residential Accounts*

  Residential subscribers of this service are guaranteed 128 kilobit per second downstream bandwidth during business hours and 256 kilobit per second at other times (for example from 6 p.m. to 8 a.m.), and a maximum data burst rate of 5 megabit per second with available excess bandwidth.

- *Silver Service for Business Accounts*

  Business accounts subscribing to this service are guaranteed 128 kilobit per second downstream data rate during business hours and 64 kilobit per second during other times, and a maximum burst rate of 1 megabit per second.

- *Silver Service for Residential Accounts*

  Subscribers are guaranteed 64 kilobit per second downstream bandwidth during business hours and 128 kilobit per second at other times, with a maximum burst rate of 1 megabit per second.

- *"On Demand" Service by Special Order*

  This class of "on demand" service allows a subscriber to request additional bandwidth available for a specific period of time. For example, a subscriber can go to operator's web site and requests for increased guaranteed bandwidth service levels from his registered subscribed class of service from the normal 256 kilobit per second to 1 megabit per second from 2 p.m. to 4 p.m. the following day only, after which his service levels returns to the original subscribed class. The provisioning server will check the bandwidth commitment and utilization history to decide whether such "on demand" service is granted.

D.3.3 Usage-Based Billing

A complete billing solution involves the following processes:

- Design different usage-based billing options
- Capture and manage subscriber account and service subscription information
- Estimate future usage based on past history
- Collect billable event data
- Generate and rate billing records
- Calculate, prepare and deliver bill
- Process and manage bill payment information and records
- Handle customer account inquires
- Manage debt and fraud

This Specification focuses only on various business scenarios on bandwidth-centric usage-based billing options.

D.3.4 Designing Usage-Based Billing Models

In support of the offering of different class of services is a new set of billing processes, which are based on the accounting of actual usage of subscribed service by each subscriber calculated by the associated fee structures.

There are several alternatives to implementing usage-based billing. The following offers a few examples:

- *Billing Based on an Average Bandwidth Usage.*

  The average bandwidth usage is defined as the total bytes transmitted divided by the billing period.

- *Billing Based on Peak Bandwidth Usage.*

  The peak bandwidth usage is the highest bandwidth usage sample during the entire billing period. Each usage sample is defined as the average bandwidth usage over a data collection period (typically 10 minutes).

Since it is usually the peak usage pattern that creates the highest possibility of access problems for the cable operator, therefore it is reasonable to charge for such usage. One scheme of peak usage billing is

called “95 percentile billing”. The process is as follows -- at the end of each billing period, the billing software examines the usage records of each subscriber and it “throws away” the top five percent of usage records of that period, then charge the subscriber on the next highest bandwidth usage.

- *“Flat Monthly Fee" Plus Usage Billing Based on the Class of Service Subscribed.*

  Any usage beyond the minimum guaranteed bandwidth for that particular subscriber service class is subject to an extra charge based on the number of bytes transmitted.

- *Billing for “On Demand” Service*

  This special billing process is to support the “On Demand” Service offering described above.

Appendix E. IPDR Standards Submission for DOCSIS 1.1 Cable Data Systems Subscriber Usage Billing Records[78]

E.1 Service Definition

Note well: This appendix is a verbatim copy of the DOCSIS Service Definition submission to IPDR.org. It is included here for information purposes only. All relevant DOCSIS requirements are contained in Section 4.1 of this specification.

Cable Data Systems consist of Cable Modem Termination Systems (CMTSs), located at a Multiple Service Operator's (MSO's) head-end office, that provide broadband Internet access to subscribers connected via Cable Modems (CMs) through the Hybrid Fiber Coax (HFC) cable plant. These Cable Data Systems comply with the Data Over Cable Service Interface Specifications (DOCSIS) sponsored by Cable Television Laboratories, Inc. The IPDR format for Cable Data Systems Subscriber Usage Billing Records specified herein support the DOCSIS 1.1 and 2.0 Operations Support System Interface specification (OSSI). The DOCSIS 1.1 and 2.0 OSSI specifications require the CMTS to provide usage-billing records for all bandwidth consumed by the subscribers connected to it via their Cable Modems when polled by the MSO's billing or mediation system.

E.1.1 DOCSIS Service Requirements

1. Cable Data Service is "always on". Thus, from the CMTS perspective, there are no subscriber logon events to track, but rather, in a manner similar to electric power utilities, there are only data traffic flows to meter and police.
2. Cable Data Subscribers are uniquely identified by their Cable Modem MAC addresses (i.e. Ethernet addresses). Note that a CM is usually assigned a dynamic IP address via DHCP, so the IP address of a subscriber may change over time. Since the CM MAC address is constant, it must be used to identify the subscriber's usage billing records. All Internet traffic generated by the subscriber's Customer Premises Equipment (CPE) is bridged by the CM to and from the CMTS. The subscriber's packet and byte (octet) traffic counts are recorded by the CMTS in Service Flow counters associated with the CM MAC address. A CM may have 2 or more Service Flows active during a collection interval. Note that the current IP addresses of the CM and all the CPE in use during the collection interval are recorded for auditing purposes.
3. Cable Data Service is metered and enforced against a Service Level Agreement (SLA) that specifies the Quality of Service (QoS) that an MSO provides to a subscriber. An MSO typically has several Service Packages to offer to their subscribers, such as "Gold", "Silver", or "Bronze". Each of the Service Packages implements a specific SLA and is available for a specific price. A Service Package is implemented by a set of Service Flows that are known to the billing system by their Service Flow IDs (SFIDs) and Service Class Names (SCNs). Service Flows are the unit of billing data collection for a Cable Data Subscriber. In addition, since a subscriber may change their Service Package over time, it is very likely that a given subscriber will have several IPDRs, one for each Service Flow they have used during the collection interval.
4. Bandwidth in a Cable Data System is measured separately in both the downstream and upstream directions (relative to the CMTS). Each Service Flow is unidirectional and may be associated with packet traffic of a specific type (e.g. TCP or UDP). Since most SLAs provide for asymmetric bandwidth guarantees, it is necessary to separate the downstream and upstream traffic flows in the billing usage records. Bandwidth used is measured in both packets and octets. If the CM is registered in DOCSIS 1.0 mode, then there will be a pair of Service Flows that contain the aggregate packet or octet count for the DOCSIS 1.0 service in each direction.

[78] Replace Appendix E per ECN OSS-N-02197 by GO on 11/15/02.

5. The bandwidth guarantee component of the SLA is enforced and metered by the CMTS with the assistance of the CM. However, the CM is not considered a trusted device because of its location on the Customer's Premises, so the CMTS is expected to provide all of the usage billing information for each subscriber connected to it.
6. Since an SLA may require the CMTS to enforce bandwidth limits by dropping or delaying packets that exceed the maximum throughput bandwidth for a Service Flow, the SLA dropped packets counters and delayed packets counters are also included in the usage records for each Service Flow. These counters are not intended to compute billable subscriber usage but rather are available to the billing and customer care systems to enable "up-selling" to subscribers who consistently exceed their subscribed service level. Thus, subscribers whose usage patterns indicate a large number of dropped octets are probably candidates for an upgrade to a higher SLA that supports their true application bandwidth demands which, in turn, generates more revenue for the MSO.
7. The packet and octet values in the usage billing records are based on absolute 64-bit counters maintained in the CMTS. These counters may be reset when the CMTS system resets, therefore the CMTS system up time (CMTSsysUpTime) is included in the IPDRdoc so that the billing or mediation system can correlate counters that appear to regress.
8.

E.1.2 DOCSIS IPDR Service Usage Element List

A DOCSIS IPDR is constructed from a number of elements that describe the IPDR itself, the CMTS that is serving the subscriber, the subscriber's CM and CPE, and the service flow attributes and counters. See the DOCSIS-3.1-B.0.xsd schema in section E.2.1 and the summary table (Table 1) below.

E.1.2.1 IPDR Information

E.1.2.1.1 IPDRcreationTime

A generic IPDR allows for an optional IPDRcreationTime element. This element is required in DOCSIS IPDRs and is formatted the same as the IPDRDoc creationTime attribute. The IPDRcreationTime is the same as the IPDRDoc creationTime when the SFtype is Interim (i.e. running service flows), but is the time the service flow was deleted when the SFtype is Stop (i.e. terminated service flows). Note that the time zone is always GMT for DOCSIS IPDRs.

E.1.2.1.2 seqNum

The optional seqNum element is not required in DOCSIS IPDRs and MUST NOT be present.

E.1.2.2 CMTS Information

A DOCSIS IPDR contains the following elements that identify the CMTS that is serving the subscriber. Each IPDR within the IPDRDoc will contain identical values for these elements since all the IPDRs are based on information maintained by the same CMTS.

E.1.2.2.1 CMTShostName

CMTShostName is the fully qualified domain name (FQDN) of the CMTS. This element is required and will be null only if the CMTS does not have a domain name. A null FQDN will be represented as <CMTShostName></CMTShostName > or < CMTShostName />.

E.1.2.2.2 CMTSipAddress

CMTSipAddress is the IPv4 address for the CMTS. This element is formatted in standard decimal dotted notation such as 10.10.100.1. This element is required.

E.1.2.2.3 CMTSsysUpTime

The sysUpTime value taken from the CMTS at the time the IPDRDoc is created formatted in decimal notation. This value does not change within an IPDRDoc. This is the number of 100ths of a second since the CMTS management interface was initialized. This value is used to determine if the CMTS was reinitialized between IPDRDoc files. In this case, the values of the service flow counters between adjacent IPDRDoc files will appear to regress. This element is required.

E.1.2.3 Subscriber Information

A DOCSIS IPDR contains the following elements that uniquely identify the subscriber. Each IPDR for a given subscriber within the IPDRDoc will contain identical values for these elements.

E.1.2.3.1 SubscriberId

The subscriber is uniquely identified by the CM's MAC address. This is the ethernet address of the cable side of the CM formatted in dashed hex notation such as 11-11-11-11-11-11. This element is required.

E.1.2.3.2 CmdocsisMode

A CM may register in one of several DOCSIS version compatibility modes. The DOCSIS mode may be specified as "1.0", "1.1", or "2.0". This element is required

E.1.2.3.3 CMipAddress

The CM is always assigned an IPv4 address on the cable side so that it can be managed via SNMP. This is the IP address assigned by DHCP when the CM last registered with the CMTS. This element is formatted in standard decimal dotted notation such as 10.101.1.123. Note that this address is dynamic and may be different between adjacent IPDRDoc files. This element is required.

E.1.2.3.4 CPEipAddress

The IPv4 address assigned to each CPE using this CM during the reporting interval. This is a comma-separated list of IPv4 addresses or null. This element is formatted in standard decimal dotted notation such as 12.12.1.121 or 12.12.12.123, 12.12.12.124, 12.12.12.125. If the CMTS is not tracking CPE IP addresses, then this element will be null (i.e. <CPEipAddress></CPEipAddress> or <CPEipAddress/>). This element is required.

E.1.2.4 Service Flow Information

A DOCSIS IPDR contains the following elements that identify the service flow and contain the counters maintained by the CMTS for that service flow.

E.1.2.4.1 SFtype

The service flow type may be either *Interim* or *Stop*. An Interim type indicates a running service flow. A Stop type indicates a terminated service flow. A terminated service flow is only reported once in the IPDRDoc that is created on the cycle after the service flow is deleted. An Interim service flow is reported in each IPDRDoc that is created while it is running. This element is required.

E.1.2.4.2 SFID

A service flow is known internally to the CMTS by its Service Flow Id (SFID) relative to its cable MAC interface. This is a unique identifier composed of two components: 1) the cable MAC layer interface id and 2) the SFID relative to the cable MAC layer interface. This is represented as ifIndex.SFID in decimal notation. Note that ifIndex is the internal interface index maintained by the SNMP agent in the CMTS for

the cable MAC layer interface that is serving the CM. This value can be used to correlate service flow counters between adjacent IPDRDoc files. To prevent confusion in the billing system, the CMTS is required to not reuse the SFID component for a minimum of 2 collection cycles. This element is required.

E.1.2.4.3 serviceClassName

This is the Service Class Name (SCN) that is assigned to this service flow by the CMTS. This is the external name associated with a QoS parameter set in the CMTS. The QoS parameter set defines how to treat the packets within a service flow for SLA enforcement purposes. Examples names might be GoldUp, GoldDn, SilverUp, SilverDn, PrimaryUp, PrimaryDn, etc. Note that the use of an SCN within the DOCSIS cable interface between the CM and the CMTS is optional, but for billing purposes, it is highly recommended. This element, however, is required within a DOCSIS IPDR and if there is no SCN assigned by the CMTS, then the value of this element is null (i.e. <serviceClassName></serviceClassName> or <serviceClassName/>. Note also that when a CM registers in DOCSIS 1.0 mode there will be no SCNs assigned and this element will be null.

E.1.2.4.4 SFdirection

The service flow direction is either *Upstream* or *Downstream* relative to the CMTS cable interface. This element is required.

E.1.2.4.5 octetsPassed

The current (or final count) of octets passed by this service flow. This is in decimal notation and is based on a 64-bit counter value maintained in the CMTS. This counter value will not overflow within the service lifetime of the CMTS. This element is required. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate octets passed count for the DOCSIS 1.0 service in this direction.

E.1.2.4.6 pktsPassed

The current (or final count) of packets passed by this service flow. This is in decimal notation and is based on a 64-bit counter value maintained in the CMTS. This counter value will not overflow within the service lifetime of the CMTS. This element is required. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate packets passed count for the DOCSIS 1.0 service in this direction.

E.1.2.4.7 SLAdropPkts

The current (or final count) of packets dropped by this service flow when enforcing the maximum throughput for this SLA (as implemented by the QoS parameter set for this service flow). This is in decimal notation and is based on a 64-bit counter value maintained in the CMTS. This counter value will not overflow within the service lifetime of the CMTS. This element is required for all service flows. Note that this value is the count of packets dropped by the CMTS for upstream service flows. Upstream packets dropped by the CM are not counted here. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate SLA drop packet count for the DOCSIS 1.0 service in this direction.

E.1.2.4.8 SLAdelayPkts

The current (or final count) of packets delayed by this service flow when enforcing the maximum throughput for this SLA (as implemented by the QoS parameter set for this service flow). This is in decimal notation and is based on a 64-bit counter value maintained in the CMTS. This counter value will not overflow within the service lifetime of the CMTS. This element is required for all service flows. Note that this value is the count of packets delayed by the CMTS for upstream service flows. Upstream packets delayed by the CM are not counted here. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate SLA packet delay count for the DOCSIS 1.0 service in this direction.

Table 33. Service Usage Element Names

Category	Name	Type	Presence	Possible Values	Remarks
CMTS Information					
Where	CMTShostName	String	Required	e.g. cmts01.mso.com	CMTS's fully qualified domain name (FQDN), if given or null
Where	CMTSipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	CMTS's IPv4 address. Canonical IP address in dotted decimal notation.
When	CMTSsysUpTime	unsignedInt	Required	nnnnnnnnn	32-bit count of hundredths of a second since CMTS system initialization, in decimal notation.
Subscriber Information					
Who	subscriberId	String	Required	hh-hh-hh-hh-hh-hh	Subscriber identified by the Cable Modem MAC address in dash delimited hex notation.
Who	CMdocsisMode	String	Required	1.0 | 1.1 | 2.0	CM's DOCSIS registration mode.
Who	CMipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	CM's current IPv4 address. Canonical IP address in dotted decimal notation.
Who	CPEipAddress	String	Required	nnn.nnn.nnn.nnn, nnn.nnn.nnn.nnn, nnn.nnn.nnn.nnn …	Current IPv4 address of all CPE using this CM, if any, or null. Comma separated list of IPv4 addresses in dotted decimal notation. One element for all CPE active during the collection interval.
Service Flow Information					
What	SFtype	String	Required	Interim | Stop	Interim identifies running Service Flows (SFs). Stop identifies terminated SFs.
What	SFID	String	Required	<ifIndex>.<SFID> e.g. 17.123456	Service Flow ID of the SF relative to its RFI MAC layer interface, in dotted decimal notation.
What	serviceClassName	String.	Required	e.g. GoldDn, GoldUp, SilverDn, SilverUp	Service Class Name (SCN) of the Service Flow
What	SFdirection	String	Required	Downstream | Upstream	Direction of the SF from the CMTS cable interface
What	octetsPassed	unsignedLong	Required	64-bit counter, in decimal notation	64-bit absolute counter value of octets passed by

Category	Name	Type	Presence	Possible Values	Remarks
					this SF
What	pktsPassed	unsignedLong	Required	64-bit counter, in decimal notation	64-bit absolute counter value of packets passed by this SF
What	SLADropPkts	unsignedLong	Required	64-bit counter, in decimal notation	64-bit absolute counter value of packets dropped exceeding SLA by this SF (Upstream counters recorded at the CMTS only.)
What	SLADelayPkts	unsignedLong	Required	64-bit counter, in decimal notation	64-bit absolute counter value of packets delayed exceeding SLA by this SF (Upstream counters recorded at the CMTS only.)
Category	**Name**	**Type**	**Presence**	**Possible Values**	**Remarks**
CMTS Information					
Where	CMTShostName	String	Required	e.g. cmts01.mso.com	CMTS's fully qualified domain name (FQDN), if given or null
Where	CMTSipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	CMTS's IPv4 address. Canonical IP address in dotted decimal notation.
When	CMTSsysUpTime	unsignedInt	Required	nnnnnnnnn	32-bit count of hundredths of a second since CMTS system initialization, in decimal notation.
Subscriber Information					
Who	subscriberId	String	Required	hh-hh-hh-hh-hh-hh	Subscriber identified by the Cable Modem MAC address in dash delimited hex notation.
Who	CMipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	CM's current IPv4 address. Canonical IP address in dotted decimal notation.
Who	CPEipAddress	String	Required	nnn.nnn.nnn.nnn, nnn.nnn.nnn.nnn, nnn.nnn.nnn.nnn …	Current IPv4 address of all CPE using this CM, if any, or null. Comma separated list of IPv4 addresses in dotted decimal notation. One element for all CPE active during the collection interval.
Service Flow Information					
What	SFtype	String	Required	Interim | Stop	Interim identifies running Service Flows (SFs). Stop

Category	Name	Type	Presence	Possible Values	Remarks
					identifies terminated SFs.
What	SFID	String	Required	<ifIndex>.<SFID> e.g. 17.123456	Service Flow ID of the SF relative to its RFI MAC layer interface, in dotted decimal notation.
What	serviceClassName	String.	Required	e.g. GoldDn, GoldUp, SilverDn, SilverUp	Service Class Name (SCN) of the Service Flow
What	SFdirection	String	Required	Downstream | Upstream	Direction of the SF from the CMTS cable interface
What	octetsPassed	unsignedLong	Required	64-bit counter, in decimal notation	64-bit absolute counter value of octets passed by this SF
What	pktsPassed	unsignedLong	Required	64-bit counter, in decimal notation	64-bit absolute counter value of packets passed by this SF
What	SLADropPkts	unsignedLong	Conditional if SFtype is Downstream	64-bit counter, in decimal notation	64-bit absolute counter value of packets dropped exceeding SLA by this SF (Downstream only)
What	SLADelayPkts	unsignedLong	Conditional if SFtype is Downstream	64-bit counter, in decimal notation	64-bit absolute counter value of packets delayed exceeding SLA by this SF (Downstream only)

E.2 DOCSIS-3.1-B.0.xsd – DOCSIS IPDR Schema File

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.ipdr.org/namespaces/ipdr"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified"
        version="3.1">
   <include schemaLocation="http://www.ipdr.org/public/IPDRDoc3.1.xsd"/>
   <element name="CMTShostName" type="string">
         <annotation>
                  <documentation>
                        CMTS fully qualified domain name (FQDN) or null.
                  </documentation>
         </annotation>
   </element>
   <element name="CMTSipAddress" type="ipdr:ipV4Addr">
         <annotation>
                        <documentation>
                        CMTS IPv4 address. Canonical IP address in period
                        delimited decimal notation.
                  </documentation>
         </annotation>
   </element>
   <element name="CMTSsysUpTime" type="unsignedInt">
         <annotation>
                        <documentation>
                        32-bit count of hundredths of a second since system
                        initialization, in decimal notation.
                  </documentation>
         </annotation>
   </element>
   <element name="subscriberId" type="string">
         <annotation>
                        <documentation>
                        subscriber identified by Cable Modem MAC address, in
                        dash delimited hex notation.
                  </documentation>
         </annotation>
   </element>
   <element name="CMdocsisMode">
         <annotation>
                        <documentation>
```

```
                        CM current DOCSIS registration mode.
                  </documentation>
            </annotation>
            <simpleType>
                        <restriction base="string">
                              <enumeration value="1.0"/>
                              <enumeration value="1.1"/>
                              <enumeration value="2.0"/>
                        </restriction>
            </simpleType>
      </element>
      <element name="CMipAddress" type="ipdr:ipV4Addr">
            <annotation>
                        <documentation>
                        CM current IPv4 address. Canonical IP address in
                        period delimited decimal notation.
                  </documentation>
            </annotation>
      </element>
      <element name="CPEipAddress" type="string">
            <annotation>
                        <documentation>
                        Comma separated list of current CPE IPv4 addresses
                        using this CM during the collection interval or null.
                  </documentation>
            </annotation>
      </element>
      <element name="serviceClassName" type="string">
            <annotation>
                        <documentation>
                        Service Class Name (SCN) of the Service Flow or
                        null.
                  </documentation>
            </annotation>
      </element>
      <element name="SFdirection">
            <annotation>
                        <documentation>
                        Direction of the SF from the CMTS cable
                        interface.
                  </documentation>
            </annotation>
            <simpleType>
                        <restriction base="string">
                              <enumeration value="Downstream"/>
```

```
                        <enumeration value="Upstream"/>
                    </restriction>
        </simpleType>
    </element>
    <element name="SFtype">
        <annotation>
                    <documentation>
                    "Interim" identifies running SFs. "Stop" identifies
                    deleted SFs.
                </documentation>
        </annotation>
        <simpleType>
                    <restriction base="string">
                        <enumeration value="Interim"/>
                        <enumeration value="Stop"/>
                    </restriction>
        </simpleType>
    </element>
    <element name="SFID" type="string">
        <annotation>
                    <documentation>
                    Service Flow ID including its RFI MAC interface
                    identifier. Formatted as "ifIndex.SFID" in decimal
                    notation.
                </documentation>
        </annotation>
    </element>
    <element name="octetsPassed" type="unsignedLong">
        <annotation>
                    <documentation>
                    64-bit absolute counter value of octets passed by
                    this SF.
                </documentation>
        </annotation>
    </element>
    <element name="pktsPassed" type="unsignedLong">
        <annotation>
                    <documentation>
                    64-bit absolute counter value of packets passed by
                    this SF.
                </documentation>
        </annotation>
    </element>
    <element name="SLAdropPkts" type="unsignedLong">
        <annotation>
```

```
                        <documentation>
                        64-bit absolute counter value of packets dropped
                        exceeding SLA by this SF (Upstream is CMTS-side
                        counter only).
                  </documentation>
            </annotation>
      </element>
      <element name="SLAdelayPkts" type="unsignedLong">
            <annotation>
                        <documentation>
                        64-bit absolute counter value of packets delayed
                        exceeding SLA by this SF (Upstream is CMTS-side
                        counter only).
                  </documentation>
            </annotation>
      </element>
      <complexType name="DOCSIS-Type">
            <complexContent>
                        <extension base="ipdr:IPDRType">
                              <sequence>
                                    <element ref="ipdr:CMTShostName"/>
                                    <element ref="ipdr:CMTSipAddress"/>
                                    <element ref="ipdr:CMTSsysUpTime"/>
                                    <element ref="ipdr:subscriberId"/>
                                    <element ref="ipdr:CMdocsisMode"/>
                                    <element ref="ipdr:CMipAddress"/>
                                    <element ref="ipdr:CPEipAddress"/>
                                    <element ref="ipdr:SFtype"/>
                                    <element ref="ipdr:SFID"/>
                                    <element ref="ipdr:serviceClassName"/>
                                    <element ref="ipdr:SFdirection"/>
                                    <element ref="ipdr:octetsPassed"/>
                                    <element ref="ipdr:pktsPassed"/>
                                    <element ref="ipdr:SLAdropPkts"/>
                                    <element ref="ipdr:SLAdelayPkts"/>
                              </sequence>
                        </extension>
            </complexContent>
      </complexType>
</schema>
```

E.3 Example IPDRDoc XML File Containing DOCSIS Subscriber Usage IPDRs

```
<?xml version="1.0" encoding="UTF-8"?>
<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="DOCSIS-3.1-B.0.xsd"
docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
creationTime="2002-06-12T21:16:23Z"
IPDRRecorderInfo="cmts01.mso.com"
version="3.1">
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:11:51Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
      <CMTSipAddress>10.10.10.124</CMTSipAddress>
      <CMTSsysUpTime>15692100</CMTSsysUpTime>
      <subscriberId>11-11-11-11-11-11</subscriberId>
      <CMdocsisMode>1.1</CMdocsisMode>
      <CMipAddress>10.70.31.252</CMipAddress>
      <CPEipAddress>1.1.4.1</CPEipAddress>
      <SFtype>Stop</SFtype>
      <SFID>16.2323</SFID>
      <serviceClassName>PrimaryUp</serviceClassName>
      <SFdirection>Upstream</SFdirection>
      <octetsPassed>108</octetsPassed>
      <pktsPassed>1</pktsPassed>
      <SLAdropPkts>0</SLAdropPkts>
      <SLAdelayPkts>0</SLAdelayPkts>
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:11:51Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
      <CMTSipAddress>10.10.10.124</CMTSipAddress>
      <CMTSsysUpTime>15692100</CMTSsysUpTime>
      <subscriberId>11-11-11-11-11-11</subscriberId>
      <CMdocsisMode>1.1</CMdocsisMode>
      <CMipAddress>10.70.31.252</CMipAddress>
      <CPEipAddress>1.1.4.1</CPEipAddress>
      <SFtype>Stop</SFtype>
      <SFID>16.2324</SFID>
      <serviceClassName>PrimaryDn</serviceClassName>
      <SFdirection>Downstream</SFdirection>
      <octetsPassed>108</octetsPassed>
      <pktsPassed>1</pktsPassed>
      <SLAdropPkts>0</SLAdropPkts>
      <SLAdelayPkts>0</SLAdelayPkts>
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:11:51Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
      <CMTSipAddress>10.10.10.124</CMTSipAddress>
      <CMTSsysUpTime>15692100</CMTSsysUpTime>
      <subscriberId>11-11-11-11-11-11</subscriberId>
      <CMdocsisMode>1.1</CMdocsisMode>
      <CMipAddress>10.70.31.252</CMipAddress>
      <CPEipAddress>1.1.4.1</CPEipAddress>
      <SFtype>Stop</SFtype>
      <SFID>16.2325</SFID>
      <serviceClassName>SilverUp</serviceClassName>
      <SFdirection>Upstream</SFdirection>
      <octetsPassed>6930432</octetsPassed>
      <pktsPassed>12995</pktsPassed>
      <SLAdropPkts>0</SLAdropPkts>
      <SLAdelayPkts>0</SLAdelayPkts>
```

```
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:11:51Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
      <CMTSipAddress>10.10.10.124</CMTSipAddress>
      <CMTSsysUpTime>15692100</CMTSsysUpTime>
      <subscriberId>11-11-11-11-11-11</subscriberId>
      <CMdocsisMode>1.1</CMdocsisMode>
      <CMipAddress>10.70.31.252</CMipAddress>
      <CPEipAddress>1.1.4.1</CPEipAddress>
      <SFtype>Stop</SFtype>
      <SFID>16.2326</SFID>
      <serviceClassName>SilverDn</serviceClassName>
      <SFdirection>Downstream</SFdirection>
      <octetsPassed>22002176</octetsPassed>
      <pktsPassed>42973</pktsPassed>
      <SLAdropPkts>0</SLAdropPkts>
      <SLAdelayPkts>0</SLAdelayPkts>
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
      <CMTSipAddress>10.10.10.124</CMTSipAddress>
      <CMTSsysUpTime>15692100</CMTSsysUpTime>
      <subscriberId>11-11-11-11-11-11</subscriberId>
      <CMdocsisMode>1.1</CMdocsisMode>
      <CMipAddress>10.70.31.252</CMipAddress>
      <CPEipAddress>1.1.4.1</CPEipAddress>
      <SFtype>Interim</SFtype>
      <SFID>16.2335</SFID>
      <serviceClassName>PrimaryUp</serviceClassName>
      <SFdirection>Upstream</SFdirection>
      <octetsPassed>0</octetsPassed>
      <pktsPassed>0</pktsPassed>
      <SLAdropPkts>0</SLAdropPkts>
      <SLAdelayPkts>0</SLAdelayPkts>
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
      <CMTSipAddress>10.10.10.124</CMTSipAddress>
      <CMTSsysUpTime>15692100</CMTSsysUpTime>
      <subscriberId>11-11-11-11-11-11</subscriberId>
      <CMdocsisMode>1.1</CMdocsisMode>
      <CMipAddress>10.70.31.252</CMipAddress>
      <CPEipAddress>1.1.4.1</CPEipAddress>
      <SFtype>Interim</SFtype>
      <SFID>16.2336</SFID>
      <serviceClassName>PrimaryDn</serviceClassName>
      <SFdirection>Downstream</SFdirection>
      <octetsPassed>0</octetsPassed>
      <pktsPassed>0</pktsPassed>
      <SLAdropPkts>0</SLAdropPkts>
      <SLAdelayPkts>0</SLAdelayPkts>
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
      <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
      <CMTShostName>cmts01.mso.com</CMTShostName>
```

```
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>11-11-11-11-11-11</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
        <CMipAddress>10.70.31.252</CMipAddress>
        <CPEipAddress>1.1.4.1</CPEipAddress>
        <SFtype>Interim</SFtype>
        <SFID>16.2337</SFID>
        <serviceClassName>SilverUp</serviceClassName>
        <SFdirection>Upstream</SFdirection>
        <octetsPassed>2108416</octetsPassed>
        <pktsPassed>3664</pktsPassed>
        <SLAdropPkts>0</SLAdropPkts>
        <SLAdelayPkts>0</SLAdelayPkts>
    </IPDR>
    <IPDR xsi:type="DOCSIS-Type">
        <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>11-11-11-11-11-11</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
        <CMipAddress>10.70.31.252</CMipAddress>
        <CPEipAddress>1.1.4.1</CPEipAddress>
        <SFtype>Interim</SFtype>
        <SFID>16.2338</SFID>
        <serviceClassName>SilverDn</serviceClassName>
        <SFdirection>Downstream</SFdirection>
        <octetsPassed>6648320</octetsPassed>
        <pktsPassed>12974</pktsPassed>
        <SLAdropPkts>0</SLAdropPkts>
        <SLAdelayPkts>0</SLAdelayPkts>
    </IPDR>
    <IPDR xsi:type="DOCSIS-Type">
        <IPDRcreationTime>2002-06-12T21:11:46Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>22-22-22-22-22-22</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
        <CMipAddress>10.70.30.254</CMipAddress>
        <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
        <SFtype>Stop</SFtype>
        <SFID>16.2321</SFID>
        <serviceClassName>GoldUp</serviceClassName>
        <SFdirection>Upstream</SFdirection>
        <octetsPassed>16181248</octetsPassed>
        <pktsPassed>31604</pktsPassed>
        <SLAdropPkts>0</SLAdropPkts>
        <SLAdelayPkts>0</SLAdelayPkts>
    </IPDR>
    <IPDR xsi:type="DOCSIS-Type">
        <IPDRcreationTime>2002-06-12T21:11:46Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>22-22-22-22-22-22</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
```

```
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Stop</SFtype>
            <SFID>16.2322</SFID>
            <serviceClassName>GoldDn</serviceClassName>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>22268928</octetsPassed>
            <pktsPassed>43494</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:11:46Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>22-22-22-22-22-22</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Stop</SFtype>
            <SFID>16.2319</SFID>
            <serviceClassName>PrimaryUp</serviceClassName>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>91</octetsPassed>
            <pktsPassed>1</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:11:46Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>22-22-22-22-22-22</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Stop</SFtype>
            <SFID>16.2320</SFID>
            <serviceClassName>PrimaryDn</serviceClassName>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>91</octetsPassed>
            <pktsPassed>1</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>22-22-22-22-22-22</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2333</SFID>
```

```
            <serviceClassName>GoldUp</serviceClassName>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>4623360</octetsPassed>
            <pktsPassed>9020</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>22-22-22-22-22-22</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2334</SFID>
            <serviceClassName>GoldDn</serviceClassName>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>6939136</octetsPassed>
            <pktsPassed>13542</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>22-22-22-22-22-22</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2331</SFID>
            <serviceClassName>PrimaryUp</serviceClassName>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>0</octetsPassed>
            <pktsPassed>0</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>22-22-22-22-22-22</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.30.254</CMipAddress>
            <CPEipAddress>1.1.3.1, 1.1.3.2, 1.1.3.4</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2332</SFID>
            <serviceClassName>PrimaryDn</serviceClassName>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>0</octetsPassed>
            <pktsPassed>0</pktsPassed>
```

```
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:11:22Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>33-33-33-33-33-33</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.31.254</CMipAddress>
            <CPEipAddress>1.1.2.1</CPEipAddress>
            <SFtype>Stop</SFtype>
            <SFID>16.2315</SFID>
            <serviceClassName>PrimaryUp</serviceClassName>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>189</octetsPassed>
            <pktsPassed>2</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:11:22Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>33-33-33-33-33-33</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.31.254</CMipAddress>
            <CPEipAddress>1.1.2.1</CPEipAddress>
            <SFtype>Stop</SFtype>
            <SFID>16.2316</SFID>
            <serviceClassName>PrimaryDn</serviceClassName>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>189</octetsPassed>
            <pktsPassed>2</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:11:22Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>33-33-33-33-33-33</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.31.254</CMipAddress>
            <CPEipAddress>1.1.2.1</CPEipAddress>
            <SFtype>Stop</SFtype>
            <SFID>16.2317</SFID>
            <serviceClassName>PlatinumUp</serviceClassName>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>22837248</octetsPassed>
            <pktsPassed>44604</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
```

```
        <IPDRcreationTime>2002-06-12T21:11:22Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>33-33-33-33-33-33</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
        <CMipAddress>10.70.31.254</CMipAddress>
        <CPEipAddress>1.1.2.1</CPEipAddress>
        <SFtype>Stop</SFtype>
        <SFID>16.2318</SFID>
        <serviceClassName>PlatinumDn</serviceClassName>
        <SFdirection>Downstream</SFdirection>
        <octetsPassed>22976512</octetsPassed>
        <pktsPassed>44876</pktsPassed>
        <SLAdropPkts>0</SLAdropPkts>
        <SLAdelayPkts>0</SLAdelayPkts>
    </IPDR>
    <IPDR xsi:type="DOCSIS-Type">
        <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>33-33-33-33-33-33</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
        <CMipAddress>10.70.31.254</CMipAddress>
        <CPEipAddress>1.1.2.1</CPEipAddress>
        <SFtype>Interim</SFtype>
        <SFID>16.2327</SFID>
        <serviceClassName>PrimaryUp</serviceClassName>
        <SFdirection>Upstream</SFdirection>
        <octetsPassed>0</octetsPassed>
        <pktsPassed>0</pktsPassed>
        <SLAdropPkts>0</SLAdropPkts>
        <SLAdelayPkts>0</SLAdelayPkts>
    </IPDR>
    <IPDR xsi:type="DOCSIS-Type">
        <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
        <subscriberId>33-33-33-33-33-33</subscriberId>
        <CMdocsisMode>1.1</CMdocsisMode>
        <CMipAddress>10.70.31.254</CMipAddress>
        <CPEipAddress>1.1.2.1</CPEipAddress>
        <SFtype>Interim</SFtype>
        <SFID>16.2328</SFID>
        <serviceClassName>PrimaryDn</serviceClassName>
        <SFdirection>Downstream</SFdirection>
        <octetsPassed>0</octetsPassed>
        <pktsPassed>0</pktsPassed>
        <SLAdropPkts>0</SLAdropPkts>
        <SLAdelayPkts>0</SLAdelayPkts>
    </IPDR>
    <IPDR xsi:type="DOCSIS-Type">
        <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
        <CMTShostName>cmts01.mso.com</CMTShostName>
        <CMTSipAddress>10.10.10.124</CMTSipAddress>
        <CMTSsysUpTime>15692100</CMTSsysUpTime>
```

```
            <subscriberId>33-33-33-33-33-33</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.31.254</CMipAddress>
            <CPEipAddress>1.1.2.1</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2329</SFID>
            <serviceClassName>PlatinumUp</serviceClassName>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>7704064</octetsPassed>
            <pktsPassed>15036</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>33-33-33-33-33-33</subscriberId>
            <CMdocsisMode>1.1</CMdocsisMode>
            <CMipAddress>10.70.31.254</CMipAddress>
            <CPEipAddress>1.1.2.1</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2330</SFID>
            <serviceClassName>PlatinumDn</serviceClassName>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>7703552</octetsPassed>
            <pktsPassed>15035</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>44-44-44-44-44-44</subscriberId>
            <CMdocsisMode>1.0</CMdocsisMode>
            <CMipAddress>10.70.31.200</CMipAddress>
            <CPEipAddress>1.1.2.100</CPEipAddress>
            <SFtype>Interim</SFtype>
            <SFID>16.2401</SFID>
            <serviceClassName/>
            <SFdirection>Upstream</SFdirection>
            <octetsPassed>7704064</octetsPassed>
            <pktsPassed>15036</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDR xsi:type="DOCSIS-Type">
            <IPDRcreationTime>2002-06-12T21:16:23Z</IPDRcreationTime>
            <CMTShostName>cmts01.mso.com</CMTShostName>
            <CMTSipAddress>10.10.10.124</CMTSipAddress>
            <CMTSsysUpTime>15692100</CMTSsysUpTime>
            <subscriberId>44-44-44-44-44-44</subscriberId>
            <CMdocsisMode>1.0</CMdocsisMode>
            <CMipAddress>10.70.31.200</CMipAddress>
            <CPEipAddress>1.1.2.100</CPEipAddress>
```

```
            <SFtype>Interim</SFtype>
            <SFID>16.2402</SFID>
            <serviceClassName/>
            <SFdirection>Downstream</SFdirection>
            <octetsPassed>7703552</octetsPassed>
            <pktsPassed>15035</pktsPassed>
            <SLAdropPkts>0</SLAdropPkts>
            <SLAdelayPkts>0</SLAdelayPkts>
      </IPDR>
      <IPDRDoc.End count="26" endTime="2002-06-12T21:16:26Z"/>
</IPDRDoc>
```

Appendix F. SNMPv2c INFORM Request Definition for Subscriber Account Management (SAM)

The INFORM Request definition of account management will be specified in this section by the ECR/ECO/ECN process.

Appendix G. Summary of the CM Authentication and the Code File Authentication

The purpose of this appendix is to provide the overview of the two authentication mechanisms defined by BPI+ specification and also to provide a example of the responsibility assignment for actual operation but not to add any new requirements for the CMTS or the CM. Please refer BPI+ specification regarding the requirement for the CMTS and the CM.

G.1 Authentication of the DOCSIS 1.1 compliant CM[79]

If the CMTS is compliant to the DOCSIS 1.1/BPI+ and a DOCSIS 1.1 compliant CM is provisioned to run BPI+ by the CM configuration file, the CMTS authenticate the CM during the CM initialization by verifying the CM certificate and the manufacturer CA certificate. These certificates are contained in Auth Info message and Auth Request message separately and sent from the CM to the CMTS just after the CM registration. Only the CM with the valid certificates will be authorized by the CMTS and become ready to forward the user traffic. Note that this CM authentication won't be applied if the CMTS and/or the CM is not compliant to BPI+, or the CM is not provisioned to run BPI+.

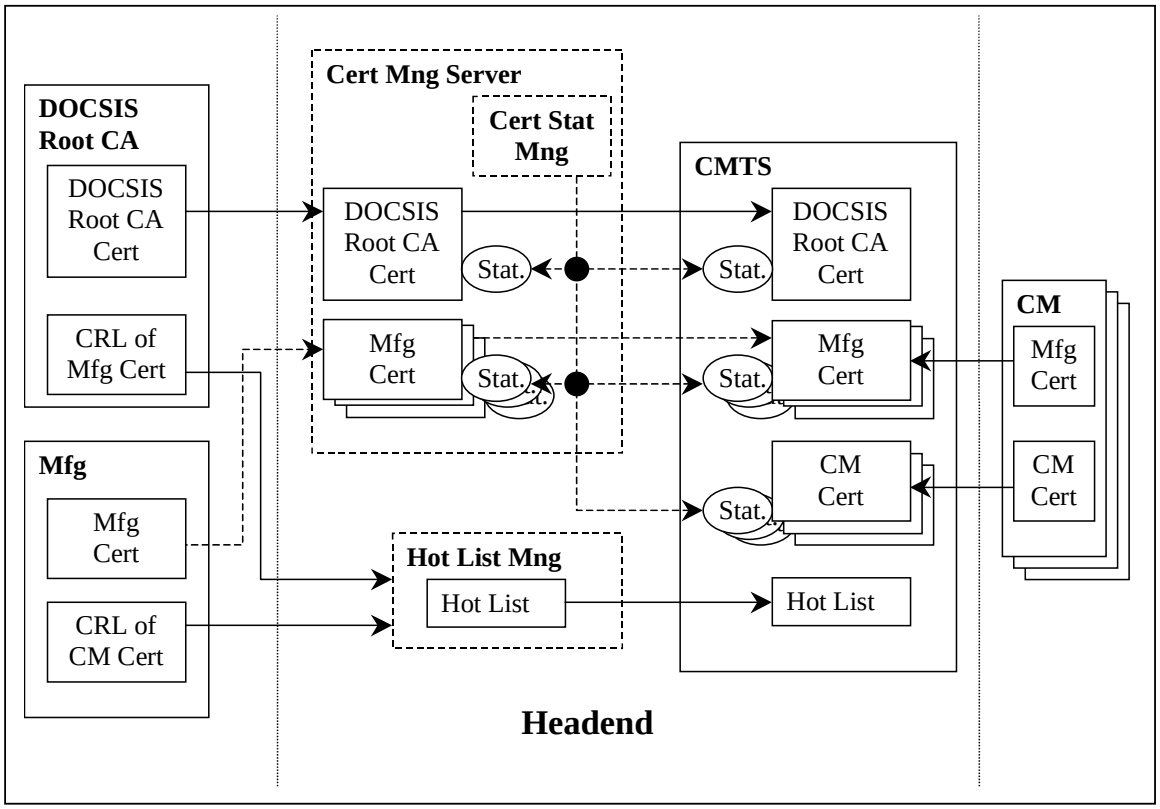

Figure 10. Authentication of the DOCSIS 1.1 compliant CM

G.1.1 Responsibility of the DOCSIS Root CA

The DOCSIS Root CA is responsible for the following:

- Store the DOCSIS Root private key in secret.
- Maintain the DOCSIS Root CA certificate.
- Issue the manufacturer CA certificates signed by the DOCSIS Root CA.

[79] Revised and/or updated various text in this section per ECN OSS-N-03066 by GO on 07/11/03.

- Maintain the CRL of the manufacturer CA.
- Provide the operators with the CRL.

It is not yet decided whether a manufacturer CA certificate signed by the DOCSIS Root CA is provided to the CM manufacturer before applying for the CableLabs' certification process or after achieving the certified status.

G.1.2 Responsibility of the CM manufacturers

The CM manufacturers are responsible for the following:

- Store the manufacturer CA private key in secret,
- Maintain the manufacturer CA certificate. The manufacturer CA certificate is usually signed by the DOCSIS Root CA but can be self-signed until the DOCSIS Root CA issues it based on the CableLabs policy.
- Issue the CM certificates,
- Put the manufacturer CA certificate in the CM's software,
- Put each CM certificate in the CM's permanent, write-once memory.
- Provide the operators with the hot list of the CM certificate. The hot list may be in the CRL format. However, the detail of the format and the way of delivery are TBD.

G.1.3 Responsibility of the operators

The operators are responsible for the following:

- Maintain that the CMTS(s) have an accurate date and time. If a CMTS has a wrong date or time, the invalid certificate may be authenticated or the valid certificate may not be authenticated.
- Put the DOCSIS Root CA certificate in the CMTS during the CMTS provisioning using DOCS-BPI2-MIB or the CMTS's proprietary function. The operator may have a server to manage this certificate for one or more CMTS(s).
- Put the manufacturer CA certificate(s) in the CMTS during the CMTS provisioning using DOCS-BPI2-MIB or the CMTS's proprietary function (optional). The operator may have a server to manage this certificate for one or more CMTS(s).
- Maintain the status of the certificates in the CMTS(s) if desired using DOCS-BPI2-MIB or the CMTS's proprietary function (optional). The operator may have a server to manage all the status of the certificates recorded in one or more CMTS(s).

The operator may have a server to manage the DOCSIS Root CA certificate, manufacturer CA certificate(s) and also the status of the certificates recorded in one or more CMTS(s).

- Maintain the hot list for the CMTS based on the CRLs provided by the DOCSIS Root CA and the CM manufacturers (optional). The operator may have a server to manage the hot list based on the CRLs provided by the DOCSIS Root CA and manufacturer CAs. The CMTS may have a function to automatically download the DOCSIS Root CA certificate and the CRLs via the Internet or other method. The DOCSIS Root CA or CableLabs is likely to put the DOCSIS Root CA on their Web or TFTP server in order to let the operators (or the CMTS on behalf of the operator) download it but this is not yet decided.

G.2 Authentication of the code file for the DOCSIS 1.1 compliant CM

When the DOCSIS 1.1/BPI+ compliant CM downloads the code file from TFTP server, the CM must always authenticate the code file as defined in the appendix D of [SP-BPI+-I03] regardless of whether the CM is provisioned to run BPI+, BPI or none of them by the CM configuration file. The CM installs the new image and restart using it only if the CVC(s) and the signature(s) in the code file are verified. If the authentication fails because of the invalid CVC(s) or signature(s) in the code file, the CM rejects the code file downloaded

from the TFTP server and continues to operate using the current code. The CM accepts the order of the software downloading via the CM configuration file or the MIB only if the CM is properly initialized by the CVC(s) in the CM configuration file. In addition to the code file authentication by the CM, the operators may authenticate the code file before they put it on the TFTP sever. The following figure shows the summary of these mechanisms.

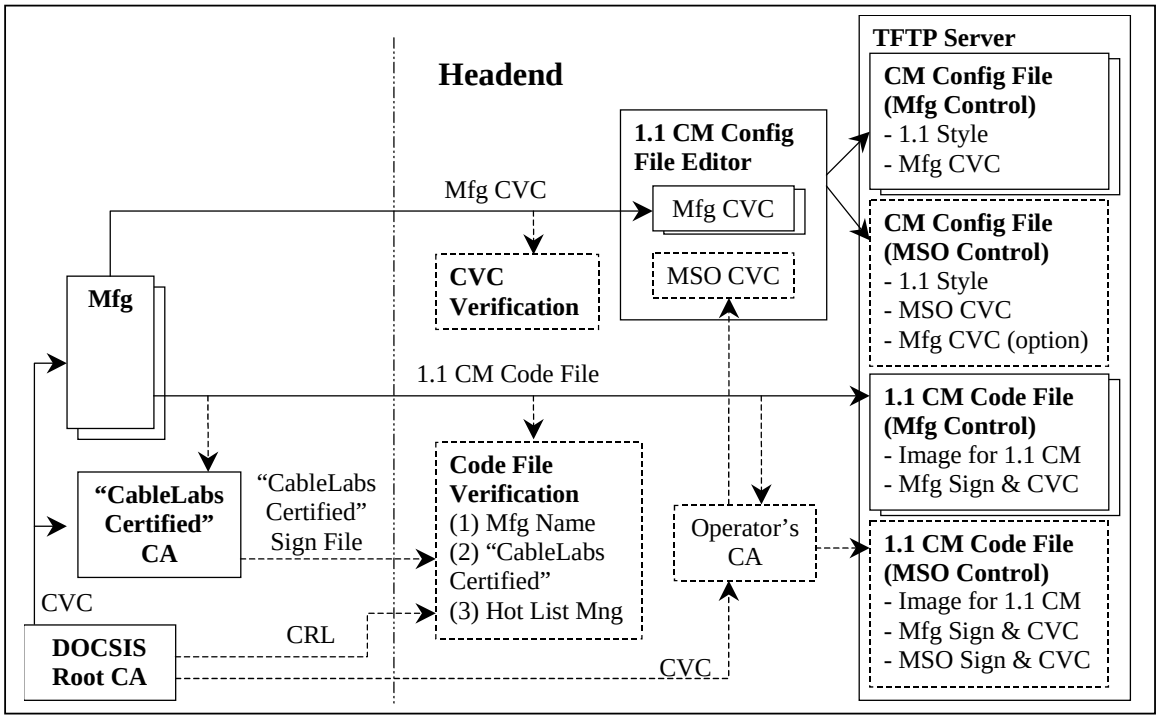

Figure 11. Authentication of the code file for the DOCSIS 1.1 compliant CM

G.2.1 Responsibility of the DOCSIS Root CA

The DOCSIS Root CA is responsible for the following:

- Store the DOCSIS Root private key in secret,
- Maintain the DOCSIS Root CA certificate, and
- Issue the code verification certificates (CVCs) for the CM manufacturers, for the operators, and for "CableLabs Certified(TM)".
- May maintain the CRL of the CVCs and provide it with the operators but not yet decided.

G.2.2 Responsibility of the CM manufacturer

The CM manufacturers are responsible for the following:

- Store the manufacturer CVC private key in secret,
- Put the DOCSIS Root CA certificate in the CM's software,
- Maintain the manufacturer CVC. (Current BPI+ specification only allows the CVC signed by the DOCSIS Root CA and does not accept the self-signed CVC.)
- Generate the code file with the manufacturer's CVC and signature, and
- Provide the operators with the code file and the manufacturer CVC,

G.2.3 Responsibility of CableLabs

CableLabs is responsible for the following:

- Store the "CableLabs Certified(TM)" CVC private key in secret,

- Maintain the "CableLabs Certified(TM)" CVC signed by the DOCSIS Root CA.
- Issue the "CableLabs Certified(TM)" signature file for the DOCSIS 1.1 CM code file certified by CableLabs.

G.2.4 Responsibility of the operators

The operator has the following responsibility and options:

- Check the manufacturer of the code file by verifying the manufacturer's CVC and signature in the code file provided by the CM manufacturer before the operator load the code file on the TFTP server (optional). The code file may be rejected and won't be loaded on the TFTP server if the unexpected manufacturer signs it or the CVC and/or the signature in it are invalid.
- Check if the code file provided by the CM manufacturer is "CableLabs Certified(TM)" by verifying the "CableLabs Certified(TM)"'s CVC and signature in the "CableLabs Certified(TM)" signature file against the code file before the operator load the code file on the TFTP server (optional). CableLabs is likely to post all the "CableLabs Certified(TM)" signature files and also the corresponding certified code files on the web or FTP server while this is not yet decided. Whether this information is open to only the CableLabs members, all the operators, all the vendors, or pubic is not yet decided
- Operate the operator CA by storing the operator CA private key in secret and maintaining the operator's (co-signer) CVC issued by the DOCSIS Root CA (optional).
- Generate the MSO-controlled code file by adding the operator's CVC and signature to the original code file provided by the CM manufacturer (optional).
- Check if the CVC provided by the CM manufacturer is valid (optional).
- Put the appropriate CVC(s) in the CM configuration file. In case that the original code file is to be downloaded to the CMs, the CM configuration file must contain the valid CVC from the CM's manufacturer. In case that the operator-controlled code file is to be downloaded, the CM configuration file must contain the valid CVC of the operator and may contain the valid CVC from the CM manufacturer. If there is no CVC in the CM configuration file or all the CVC(s) in the CM configuration file is invalid, the CM won't accept any order of the software downloading via the CM configuration file and the MIB. Note that the DOCSIS 1.1 compliant CM may be registered and authorized by the CMTS and becomes operational regardless of whether the CM configuration file contains the valid CVC(s).

Appendix H. Format and Content for Event, SYSLOG and SNMP Trap[80]

The list in this appendix summarizes the format and content for event, syslog and SNMP trap.

Please note that the list is originally derived from Appendix J of SP-RFIv1.1 “Radio Frequency Interface Specification” and is a superset of that original list. To avoid redundancy and reduce the risk of inconsistent between two documents, the Appendix J of SP-RFIv1.1 is being pointed to this list and the original list is removed from that document.

Each row specifies a possible event appears in CM or in CMTS. These events are to be reported by a cable device in any or all of the following three means: local event logging as implemented by the event table in the DOCS-CABLE-DEVICE-MIB, the syslog and the SNMP trap.

The first and second columns indicate in which stage the event happens. The third and fourth columns indicate the priority it is assigned in CM and in CMTS. These priorities are the same is reported in the docsDevEvLevel object in the DOCS-CABLE-DEVICE-MIB and in the LEVEL field of a syslog.

The Fifth column specifies the event text, which is reported in the docsDevEvText object of the DOCS-CABLE-DEVICE-MIB and the text field of the syslog. The sixth column provides additional information about the event text in the 5th column. Some of the text fields are pure English sentence. Some include variable information. The variables are explained in the sixth column. Some of the variables are only required in the SYSLOG and are described in the sixth column too. Additional vendor specific text MAY be added to the end of the event text.

The next column specifies error code set. The eighth column indicates an unique identification number for the event, which is assigned to the docsDevEvId object in the MIB and the <eventId> field of a syslog. The final column specifies the SNMP trap, which notifies this event to a SNMP event receiver.

The rules to uniquely generate an event ID from the error code are described in the section 4.4.2.2.2. Please notice that the algorithm in the section 4.4.2.2.2 will generate a hexadecimal number. The event IDs in this list are converted to decimal integers from hexadecimal number.

The syslog format is specified in the section 4.4.2.2.2 SYSLOG Message Format of this document.

The SNMP traps are defined in the cable device trap MIB.

To better illustrate the table, let us take the example of the first row in the section of DYNAMIC SERVICE REQUEST.

The first and second columns are “Dynamic Services” and “Dynamic Service Request”. The event priority is “Error” in a cable modem and “Warning” in a cable modem termination system. The event Id is 1392509184. The event text is “Service Add rejected - Unspecified reason”. The sixth column reads “For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)”. This is a note about the SYSLOG. That is to say, the syslog text body will be like “Service Add rejected - Unspecified reason - MAC addr: x1 x2 x3 x4 x5 x6”.

The last column “TRAP NAME” is docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap. That indicates that the event is notified by the SNMP trap docsDevCmDynServReqFailTrap in a cable modem and docsDevCmtsDynServReqFailTrap in a CMTS.

[80] Revised and/or updated various text in this section per ECN OSS-N-03066 by GO on 07/11/03.

Table 34. Format and Content for Event, SYSLOG and SNMP Trap[81,82,83]

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
				DOWNSTREAM ACQUISITION FAILED				
Init	DOWNSTREAMACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to acquire QAM/QPSK symbol timing		T01.0	84000100	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to acquire FEC framing		T02.0	84000200	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure, Acquired FEC framing - Failed to acquire MPEG2 Sync		T02.1	84000201	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to acquire MAC framing		T03.0	84000300	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to receive MAC SYNC frame within time-out period		T04.0	84000400	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Loss of Sync		T05.0	84000500	
				FAILED TO OBTAIN UPSTREAM PARAMETERS				
Init	OBTAIN UPSTREAM PARAMETERS	Critical		No UCDs Received - Timeout		U01.0	85000100	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		UCD invalid or channel unusable		U02.0	85000200	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		UCD & SYNC valid - NO MAPS for this channel		U04.0	85000400	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		US channel wide parameters not set before Burst Descriptors		U06.0	85000600	
				MAP Upstream Bandwidth Allocation				
Any	Any	informational	Informational	A transmit opportunity was missed because the MAP arrived too late.		M01.0	77000100	
				RANGING FAILED : RNG-REQ RANGING REQUEST				
Init	RANGING	Critical		No Maintenance Broadcasts for Ranging opportunities received - T2 time-out		R01.0	82000100	
Init	RANGING	Critical		Received Response to Broadcast Maintenance Request, But no Unicast Maintenance opportunities received - T4 timeout		R04.0	82000400	

[81] Error Code Set E206.0, E207.0, E208.0, and E209.0 updated per OSS-N-02174 by RKV on 10/24/02.

[82] Error Code Set E206.0, E207.0, and E208.0 updated per OSS-N-03009 by GO on 02/25/03.

[83] PROCESS BPKM CM PRIORITY revised per OSS-N-03005 by GO on 03/20/03.

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	RANGING		Warning	No Ranging Requests received from POLLED CM (CMTS generated polls).		R101.0	82010100	
Init	RANGING		Warning	Retries exhausted for polled CM (report MAC address). After 16 R101.0 errors.		R102.0	82010200	
Init	RANGING		Warning	Unable to Successfully Range CM (report MAC address) Retries Exhausted.	Note: this is different from R102.0 in that it was able to try, i.e. got REQs but failed to Range properly.	R103.0	82010300	
Init	RANGING		Warning	Failed to receive Periodic RNG-REQ from modem (SID X), timing-out SID.		R104.0	82010400	
				RANGING FAILED : RNG-REQ RANGING RESPONSE				
Init	RANGING	Critical		No Ranging Response received - T3 time-out		R02.0	82000200	
Init	RANGING	Critical		Ranging Request Retries exhausted		R03.0	82000300	
Init	RANGING	Critical		Started Unicast Maintenance Ranging - No Response received - T3 time-out		R05.0	82000500	
Init	RANGING	Critical		Unicast Maintenance Ranging attempted - No response - Retries exhausted		R06.0	82000600	
Init	RANGING	Critical		Unicast Ranging Received Abort Response - Re-initializing MAC		R07.0	82000700	
				TOD FAILED Before Registration				
Init	TOD	Warning		ToD request sent - No Response received		D04.1	68000401	
Init	TOD	Warning		ToD Response received - Invalid data format		D04.2	68000402	
				TOD FAILED After Registration				
TOD		Error		ToD request sent- No Response received		D04.3	68000403	docsDevCmTODFailTrap
TOD		Error		ToD Response received - Invalid data format		D04.4	68000404	docsDevCmTODFailTrap
				DHCP and TFTP FAILED - before registration				
Init	TFTP	Critical		TFTP failed - Request sent - No Response		D05.0	68000500	
Init	TFTP	Critical		TFTP failed - configuration file NOT FOUND	For SYSLOG only: append: File name = <P1> P1 = requested file name	D06.0	68000600	
Init	TFTP	Critical		TFTP Failed - OUT OF ORDER packets		D07.0	68000700	
Init	TFTP	Critical		TFTP file complete - but failed Message Integrity check MIC	For SYSLOG only: append: File name = <P1> P1 = filename of TFTP file	D08.0	68000800	
Init	TFTP	Critical		TFTP file complete – but missing mandatory TLV		D09.0	68000900	

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	TFTP	Critical		TFTP Failed - file too big		D10.0	68001000	
Init	DHCP	Critical		DHCP FAILED - Discover sent, no offer received		D01.0	68000100	
Init	DHCP	Critical		DHCP FAILED - Request sent, No response		D02.0	68000200	
Init	DHCP	Warning		DHCP WARNING - Non-critical field invalid in response		D03.0	68000300	
Init	DHCP	Critical		DHCP FAILED – Critical field invalid in response		D03.1	68000301	
				REGISTRATION FAILED (REG-REQ REGISTRATION REQUEST)				
Init	REGISTRATION REQUEST		Warning	Service unavailable - Other	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.0	73000400	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Service unavailable - Unrecognized configuration setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.1	73000401	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Service unavailable - Temporarily unavailable	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.2	73000402	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Service unavailable - Permanent	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.3	73000403	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Registration rejected authentication failure: CMTS MIC invalid	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I05.0	73000500	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	REG REQ has Invalid MAC header	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I101.0	73010100	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	REG REQ has Invalid SID or not in use	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I102.0	73010200	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	REG REQ missed Required TLV's	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I104.0	73010400	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ - Format Invalid	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I105.0	73010500	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ - Not in use	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I105.1	73010501	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ - Not Multiple of 62500 Hz	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I105.2	73010502	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US CH - Invalid or Unassigned	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I106.0	73010600	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US CH - Change followed with (RE-) Registration REQ	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I106.1	73010601	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US CH - Overload	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I107.0	73010700	docsDevCmtsInitRegReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	REGISTRATION REQUEST		Warning	Network Access has Invalid Parameter	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I108.0	73010800	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Class of Service - Invalid Configuration	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I109.0	73010900	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Class of Service - Unsupported class	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I110.0	73011000	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Class of Service - Invalid class ID or out of range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I111.0	73011100	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max DS Bit Rate - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I112.0	73011200	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max DS Bit Rate Unsupported Setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I112.1	73011201	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US Bit - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I113.0	73011300	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US Bit Rate - Unsupported Setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I113.1	73011301	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US Priority Configuration - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I114.0	73011400	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US Priority Configuration - Setting out of Range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I114.1	73011401	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I115.0	73011500	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting - Exceed Max US Bit Rate	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I115.1	73011501	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting - Out of Range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I115.2	73011502	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US CH Transmit Burst configuration setting - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I116.0	73011600	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US CH Transmit Burst configuration setting - Out of Range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I116.1	73011601	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Invalid Modem Capabilities configuration setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I117.0	73011700	docsDevCmtsInitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Configuration file contains parameter with the value outside of the range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I118.0	73011800	docsDevCmtsInitRegReqFailTrap
				VERSION 1.1 SPECIFIC REG-REQ REGISTRATION REQUEST				
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Unspecified reason	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.0	73020100	docsDevCmtsInitRegReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Unrecognized configuration setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.1	73020101	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - temporary no resource	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.2	73020102	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Permanent administrative	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.3	73020103	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Required parameter not present <P1>	P1 = TLV type It is up to the vendor to support 1 or many For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.4	73020104	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Header suppression setting not supported	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.5	73020105	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Multiple errors	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.6	73020106	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – duplicate reference-ID or index in message	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.7	73020107	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – parameter invalid for context <P1>	P1 = TLV parameter For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.8	73020108	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Authorization failure	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.9	73020109	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Major service flow error	For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.10	73020110	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Major classifier error	For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.11	73020111	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Major PHS rule error	For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.12	73020112	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Multiple major errors	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.13	73020113	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Message syntax error <P1>	P1 = message For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.14	73020114	docsDevCmtsInitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Primary service flow error <P1>	P1 = Service Flow Reference For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.15	73020115	docsDevCmtsInitRegReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Message too big <P1>	P1 = # of characters For CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.16	73020116	docsDevCmtsInitRegReqFailTrap
				REG-RSP REGISTRATION RESPONSE		I00.0	73000000	
Init	REGISTRATION RESPONSE	Critical		REG-RSP - invalid format or not recognized		I01.0	73000100	
Init	REGISTRATION RESPONSE	Critical		REG RSP not received		I02.0	73000200	
Init	REGISTRATION RESPONSE	Critical		REG RSP bad SID <P1>		I03.0	73000300	
				Version 1.1 Specific REG-RSP				
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains service flow parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.0	73025100	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains classifier parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.1	73025101	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains PHS parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.2	73025102	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected unspecified reason		I251.3	73025103	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message syntax error <P1>	P1 = message	I251.4	73025104	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message too big <P1>	P1 = # of characters	I251.5	73025105	
				REG-ACK REGISTRATION ACKNOWLEDGEMENT		I300.0	73030000	
Init	REGISTRATION ACKNOWLEDGEMENT		Warning	REG aborted no REG-ACK	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I301.0	73030100	docsDevCmtsInitRegAckFailTrap
Init	REGISTRATION Acknowledgement		Warning	REG ACK rejected unspecified reason	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I302.0	73030200	docsDevCmtsInitRegAckFailTrap
Init	REGISTRATION ACKNOWLEDGEMENT		Warning	REG ACK rejected message syntax error	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I303.0	73030300	docsDevCmtsInitRegAckFailTrap
				TLV-11 Failures		I400.0	73040000	
Init	TLV-11 PARSING	Notice		TLV-11 – unrecognized OID		I401.0	73040100	docsDevCmInitTLVUnknownTrap
Init	TLV-11 Failures	Critical		TLV-11 - Illegal Set operation failed		I402.0	73040200	
Init	TLV-11 Failures	Critical		TLV-11 – Failed to set duplicate elements		I403.0	73040300	
				SW UPGRADE INIT				
SW Upgrade	SW UPGRADE INIT	Notice		SW Download INIT - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E101.0	69010100	docsDevCmSwUpgradeInitTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
SW Upgrade	SW UPGRADE INIT	Notice		SW Download INIT - Via Config file <P1>	P1 = CM config file name For SYSLOG only, append: SW file: <P2> - SW server: < P3>. P2 = SW file name and P3 = Tftp server IP address	E102.0	69010200	docsDevCmSwUpgradeInitTrap
				SW UPGRADE GENERAL FAILURE				
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW Upgrade Failed during download - Max retry exceed (3)	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E103.0	69010300	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW Upgrade Failed Before Download - Server not Present	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E104.0	69010400	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed before download - File not Present	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E105.0	69010500	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed before download – TFTP Max Retry Exceeded	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E106.0	69010600	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed after download - Incompatible SW file	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E107.0	69010700	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed after download - SW File corruption	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E108.0	69010800	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Disruption during SW download – Power Failure	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E109.0	69010900	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Disruption during SW download - RF removed	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E110.0	69011000	docsDevCmSwUpgradeFailTrap
				SW UPGRADE SUCCESS				
SW Upgrade	SW UPGRADE SUCCESS	Notice		SW download Successful - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E111.0	69011100	docsDevCmSwUpgradeSuccessTrap
SW Upgrade	SW UPGRADE SUCCESS	Notice		SW download Successful - Via Config file	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp server IP address	E112.0	69011200	docsDevCmSwUpgradeSuccessTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
				DHCP FAILURE AFTER CM HAS REGISTERED WITH THE CMTS				
DHCP		Error		DHCP RENEW sent - No response		D101.0	68010100	docsDevCmDHCP FailTrap
DHCP		Error		DHCP REBIND sent - No response		D102.0	68010200	docsDevCmDHCP FailTrap
DHCP		Warning		DHCP RENEW WARNING – Field invalid in response		D103.0	68010300	docsDevCmDHCP FailTrap
DHCP		Critical		DHCP RENEW FAILED - Critical field invalid in response		D103.1	68010301	docsDevCmDHCP FailTrap
DHCP		Warning		DHCP REBIND WARNING – Field invalid in response		D104.0	68010400	docsDevCmDHCP FailTrap
DHCP		Critical		DHCP REBIND FAILED - Critical field invalid in response		D104.1	68010401	docsDevCmDHCP FailTrap
				DYNAMIC SERVICE REQUEST		S00		
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.0	83000100	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.1	83000101	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Temporary no resource	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.2	83000102	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Permanent administrative	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.3	83000103	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.4	83000104	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Header suppression setting not supported	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.5	83000105	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error		Service Add rejected – Service flow exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.6	83000106	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.7	83000107	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Add aborted	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.8	83000108	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.9	83000109	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier not found	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.10	83000110	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error		Service Add rejected – Classifier exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.11	83000111	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – PHS rule exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.13	83000113	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Duplicated reference-ID or index in message	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.14	83000114	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple upstream flows	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.15	83000115	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple downstream flows	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.16	83000116	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier for another flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.17	83000117	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – PHS rule for another flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.18	83000118	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Parameter invalid for context	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.19	83000119	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Authorization failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.20	83000120	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major service flow error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.21	83000121	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major classifier error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.22	83000122	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major PHS rule error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.23	83000123	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple major errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.24	83000124	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.25	83000125	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Message too big	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.26	83000126	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Temporary DCC	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.27	83000127	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.0	83000200	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.1	83000201	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Temporary no resource	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.2	83000202	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Permanent administrative	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.3	83000203	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Requester not owner of service flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.4	83000204	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Service flow not found	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.5	83000205	docsDevCmDynSerReqFailTrap, docsDevCmtsDynServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.6	83000206	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Header suppression setting not supported	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.7	83000207	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.8	83000208	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple errors	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.9	83000209	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Classifier not found	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.10	83000210	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error		Service Change rejected – Classifier exists	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.11	83000211	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule not found	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.12	83000212	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule exists	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.13	83000213	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Duplicated reference-ID or index in message	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.14	83000214	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple upstream flows	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.15	83000215	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple downstream flows	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.16	83000216	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Classifier for another flow	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.17	83000217	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule for another flow	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.18	83000218	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Invalid parameter for context	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.19	83000219	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Authorization failure	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.20	83000220	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major service flow error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.21	83000221	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected –Major classifier error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.22	83000222	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major PHS error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.23	83000223	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple major errors	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.24	83000224	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Message syntax error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.25	83000225	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Message too big	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.26	83000226	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Temporary DCC	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.27	83000227	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.0	83000300	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected –Requestor not owner of service flow	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.1	83000301	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected - Service flow not found	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.2	83000302	docsDevCmDynSe rvReqFailTrap, docsDevCmtsDyn ServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected - HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.3	83000303	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Message syntax error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.4	83000304	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
				DYNAMIC SERVICE RESPONSES				
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.0	83010100	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add aborted - No RSP	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.1	83010101	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.2	83010102	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.3	83010103	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.4	83010104	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.5	83010105	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.6	83010106	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error		Service Add Response rejected - Service Flow exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.7	83010107	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Multiple errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.8	83010108	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error		Service Add Response rejected - Classifier exists >	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.9	83010109	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - PHS rule exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.10	83010110	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Duplicate reference_ID or index in message	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.11	83010111	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Classifier for another flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.12	83010112	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Parameter invalid for context	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.13	83010113	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Major service flow error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.14	83010114	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Major classifier error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.15	83010115	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Major PHS Rule error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.16	83010116	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Multiple major errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.17	83010117	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Message too big - MAC	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.18	83010118	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.0	83010200	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change aborted- No RSP	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.1	83010201	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.2	83010202	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.4	83010204	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.5	83010205	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.6	83010206	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Multiple errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.7	83010207	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error		Service Change Response rejected – Classifier exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.8	83010208	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – PHS rule exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.9	83010209	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Duplicated reference-ID or index in	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.10	83010210	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Invalid parameter for context	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.11	83010211	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Major classifier error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.12	83010212	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Major PHS rule error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.13	83010213	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Multiple Major errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.14	83010214	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Message too big	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.15	83010215	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.3	83010203	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Delete Response rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S103.0	83010300	docsDevCmDynSe rvRspFailTrap, docsDevCmtsDyn ServRspFailTrap
				DYNAMIC SERVICE ACKNOWLEDGEME NTS				

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add Response rejected - Invalid Transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.0	83020100	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add Aborted - No ACK	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.1	83020101	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add ACK rejected - HMAC auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.2	83020102	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add ACK rejected- Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.3	83020103	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.0	83020200	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change Aborted – No ACK	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.1	83020201	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected – HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.2	83020202	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected – Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.3	83020203	docsDevCmDynServAckFailTrap, docsDevCmtsDynServAckFailTrap
				CM CONFIGURATION FILE (BPI+)				
Init (BPI+)		Error	Notice	Missing BP Configuration Setting TLV Type: <P1>	P1 = missing required TLV Type	B101.0	66010100	DocsDevCmBpiInitTrap, docsDevCmtsBpiInitTrap
Init (BPI+)		Alert	Notice	Invalid BP Configuration Setting Value: <P1> for Type: <P2>	P1=The TLV Value for P2. P2 = The first Configuration TLV Type that contain invalid value.	B102.0	66010200	docsDevCmBpiInitTrap
				AUTH FSM				
BPKM		Warning	Error	Auth Reject - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.2	66030102	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Auth Reject – Unauthorized CM	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.3	66030103	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
BPKM		Warning	Error	Auth Reject – Unauthorized SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.4	66030104	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Error	Error	Auth Reject – Permanent Authorization Failure	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.8	66030108	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Auth Reject – Time of Day not acquired	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.9	66030109	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Alert	Error	CM Certificate Error	For SYSLOG only, append: MAC addr: <P1> P1=Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.11	66030111	DocsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Auth Invalid - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.2	66030202	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Auth Invalid - Unauthorized CM	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.3	66030203	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Auth Invalid - Unsolicited	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.5	66030205	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Auth Invalid - Invalid Key Sequence Number	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.6	66030206	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Auth Invalid - Message (Key Request) Authentication Failure	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.7	66030207	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Unsupported Crypto Suite	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B303.0	66030300	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
				EVENT BETWEEN AUTH & TEK FSM				
BPKM		Informational		Authorized	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B401.0	66040100	docsDevCmBPKM Trap
BPKM		Informational		Auto Pend	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B402.0	66040200	docsDevCmBPKM Trap
BPKM		Informational		Auth Comp	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B403.0	66040300	docsDevCmBPKM Trap
BPKM		Informational		Stop	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B404.0	66040400	docsDevCmBPKM Trap
				TEK FSM				

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
BPKM		Warning	Error	Key Reject - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B501.2	66050102	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	Key Reject - Unauthorized SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B501.3	66050103	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	TEK Invalid - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B502.3	66050203	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
BPKM		Warning	Error	TEK Invalid - Invalid Key Sequence Number	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B502.6	66050206	docsDevCmBPKM Trap, docsDevCmtsBPK MTrap
				SA MAP FSM				
Dynamic SA		Informational		SA Map State Machine Started	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B601.0	66060100	docsDevCmDyna micSATrap
Dynamic SA		Warning	Error	Unsupported Crypto Suite	For SYSLOG only, append: MAC addr: <P1>. P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B602.0	66060200	docsDevCmDyna micSATrap, docsDevCmtsDyna micSATrap
Dynamic SA		Error		Map Request Retry Timeout	For CM SYSLOG only append: MAC addr: <P1>. P1 = Mac Addr of CMTS	B603.0	66060300	docsDevCmDyna micSATrap
Dynamic SA		Informational		Unmap	For CM SYSLOG only append: MAC addr: <P1>. P1 = Mac Addr of CMTS	B604.0	66060400	docsDevCmDyna micSATrap
Dynamic SA		Warning	Error	Map Reject - Not Authorized for Requested Downstream Traffic Flow (EC=7)	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B605.9	66060509	docsDevCmDyna micSATrap, docsDevCmtsDyna micSATrap
Dynamic SA		Warning	Error	Map Reject - Downstream Traffic Flow Not Mapped to BPI+ SAID (EC=8)	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B605.10	66060510	docsDevCmDyna micSATrap, docsDevCmtsDyna micSATrap
Dynamic SA		Warning	Error	Mapped to Existing SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B606.0	66060600	docsDevCmDyna micSATrap, docsDevCmtsDyna micSATrap
Dynamic SA		Warning	Error	Mapped to New SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B607.0	66060700	docsDevCmDyna micSATrap, docsDevCmtsDyna micSATrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
				VERIFICATION OF CODE FILE				
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Improper Code File Controls	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E201.0	69020100	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Manufacturer CVC Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E202.0	69020200	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Manufacturer CVS Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E203.0	69020300	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Co-Signer CVC Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E204.0	69020400	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Co-Signer CVS Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E205.0	69020500	docsDevCmSwUpgradeFailTrap
				VERIFICATION OF CVC				
SW Upgrade	VERIFICATION OF CVC	Error		Improper Configuration File CVC Format	For SYSLOG only, append: Config File: <P1> - TFTP Server: <P2> P1 = Config File Name P2 = TFTP Server IP Address	E206.0	69020600	docsDevCmSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC	Error		Configuration File CVC Validation Failure	For SYSLOG only, append: Config File: <P1> - TFTP Server: <P2> P1 = Config File Name P2 = TFTP Server IP Address	E207.0	69020700	docsDevCmSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC	Error		Improper SNMP CVC Format	For SYSLOG only, append: SNMP Manager: <P1>. P1= IP Address of SNMP Manager	E208.0	69020800	docsDevCmSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC*	Error		SNMP CVC Validation Failure	For SYSLOG only, append: SNMP Manager: <P1>. P1=IP Address of SNMP manager	E209.0	69020900	docsDevCmSwUpgradeCVCFailTrap
				UCC-REQ Upstream Channel Change Request				
UCC	UCC Request	Error	Warning	UCC-REQ received with invalid or out of range US channel ID.		C01.0	67000100	
UCC	UCC Request	Error	Warning	UCC-REQ received, unable to send UCC-RSP.		C02.0	67000200	

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
				UCC-RSP Upstream Channel Change Response				
UCC	UCC Response		Warning	UCC-RSP not received on previous channel ID.		C101.0	67010100	
UCC	UCC Response		Warning	UCC-RSP received with invalid channel ID.		C102.0	67010200	
UCC	UCC Response		Warning	UCC-RSP received with invalid channel ID on new channel.		C103.0	67010300	
				Dynamic Channel Change Request				
DCC	DCC Request	Error	Warning	DCC rejected already there		C201.0	67020100	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Informational	Notice	DCC depart old		C202.0	67020200	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Informational	Notice	DCC arrive new		C203.0	67020300	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Critical	Warning	DCC aborted unable to acquire new downstream channel		C204.0	67020400	
DCC	DCC Request	Critical	Warning	DCC aborted no UCD for new upstream channel		C205.0	67020500	
DCC	DCC Request	Critical	Warning	DCC aborted unable to communicate on new upstream channel		C206.0	67020600	
DCC	DCC Request	Error	Warning	DCC rejected unspecified reason		C207.0	67020700	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected permanent - DCC not supported		C208.0	67020800	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected service flow not found		C209.0	67020900	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected required parameter not present		C210.0	67021000	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected authentication failure		C211.0	67021100	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected multiple errors		C212.0	67021200	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected duplicate SF reference-ID or index in message		C215.0	67021500	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected parameter invalid for context		C216.0	67021600	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected message syntax error		C217.0	67021700	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIOR-ITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DCC	DCC Request	Error	Warning	DCC rejected message too big		C218.0	67021800	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
				Dynamic Channel Change Response				
DCC	DCC Response		Warning	DCC-RSP not received on old channel		C301.0	67030100	DocsDevCmDccRspFailTrap, docsDevCmtsDccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP not received on new channel		C302.0	67030200	DocsDevCmDccRspFailTrap, docsDevCmtsDccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected unspecified reason		C303.0	67030300	DocsDevCmDccRspFailTrap, docsDevCmtsDccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected unknown transaction ID		C304.0	67030400	DocsDevCmDccRspFailTrap, docsDevCmtsDccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected authentication failure		C305.0	67030500	DocsDevCmDccRspFailTrap, docsDevCmtsDccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected message syntax error		C306.0	67030600	DocsDevCmDccRspFailTrap, docsDevCmtsDccRspFailTrap
				Dynamic Channel Change Acknowledgement		C400.0		
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK not received		C401.0	67040100	DocsDevCmDccAckFailTrap, docsDevCmtsDccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected unspecified reason		C402.0	67040200	DocsDevCmDccAckFailTrap, docsDevCmtsDccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected unknown transaction ID		C403.0	67040300	DocsDevCmDccAckFailTrap, docsDevCmtsDccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected authentication failure		C404.0	67040400	DocsDevCmDccAckFailTrap, docsDevCmtsDccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected message syntax error		C405.0	67040500	DocsDevCmDccAckFailTrap, docsDevCmtsDccAckFailTrap

Appendix I. Trap Definitions for Cable Device

The trap definition for cable device will be specified in this section by the ECR/ECO/ECN process.

Appendix J. Application of RFC-2933 to DOCSIS 1.1 active/passive IGMP devices

J.1 DOCSIS 1.1 IGMP MIBs

DOCSIS 1.1 devices, CM and CMTS, that support IGMP (in active or passive mode), MUST support the IDMR IGMP MIB (RFC-2933). As such, this section describes the application of the IETF IDMR sub-committee IGMP MIB to DOCSIS 1.1 active/passive IGMP devices.

The IDMR IGMP MIB is organized into two distinct tables, the interface and cache tables. The IGMP Interface Table contains entries for each interface that supports IGMP on a device. For DOCSIS 1.1 this includes the NSI and HFC for the CMTS and the HFC and CMCI on the CM. The IGMP Cache Table contains one row for each IP Multicast Group for which there are active members on a given interface. Active membership MUST only exist on the CMCI of a Cable Modem. However, active membership MAY exist on both the NSI and HFC side interfaces of the CMTS. This is because a CMTS may be implemented as a Multicast Router on which other network side devices are actively participating in a multicast session.

Support of the IDMR IGMP MIB by DOCSIS 1.1 devices is presented in terms of IGMP capabilities, the device type (CM or CMTS), and the interface on which IGMP is supported. This is followed by a set of new IGMP MIB conformance, compliance and group statements for DOCSIS 1.1 devices.

J.1.1 IGMP Capabilities: Active and Passive Mode

There are two basic modes of IGMP capability that are applicable to a DOCSIS 1.1 device. The first mode is a *passive* operation in which the device selectively forwards IGMP based upon the known state of multicast session activity on the subscriber side (an example of this is described in Appendix L of [DOCSIS 5]). In *passive* mode, the device derives its IGMP timers based on the rules specified in section 3.3.1 of the RFI. The second mode is an *active* operation in which the device terminates and initiates IGMP based upon the known state of multicast session activity on the subscriber side. One example of the latter, active, mode is commonly referred to as an IGMP-Proxy implementation side (as described in [ID-IGMP]). A more complete example of an active IGMP device is that of a Multicast Router. Although a specific implementation is not imposed by the DOCSIS 1.1 specification, the device MUST meet the requirements stated in section 3.3.1 of [DOCSIS 5] and MUST support the IDMR IGMP MIB as described herein. As presently specified in the DOCSIS 1.1, active CMs are explicitly prohibited from transmitting IGMP Queries upstream onto the HFC. However, active CMTSs may transmit IGMP Queries onto the NSI as mentioned previously.

J.1.2 IGMP Interfaces

A description of the application of the IDMR IGMP MIB to DOCSIS 1.1 devices follows. This description is organized by CM and CMTS device type.

J.2 DOCSIS 1.1 CM Support for the IGMP MIB

There are two types of interfaces applicable to IGMP on the DOCSIS 1.1 CM. These are the HFC-Side and CMCI-Side interfaces, respectively. Application of the IGMP MIB to DOCSIS 1.1 CMs is presented in terms of *passive* and *active* CM operation and these two interface types.

J.2.1 igmpInterfaceTable- igmpInterfaceEntry

J.2.1.1 igmpInterfaceIfIndex

The ifIndex value of the interface for which IGMP is enabled.

J.2.1.1.1 All Modes

This is the same for passive and active modes.

HFC-side: not-accessible. ifIndex of docsCableMaclayer(127), CATV MAC Layer

CMCI-side: not-accessible. ifIndex of CMCI-Side interface.

J.2.1.2 igmpInterfaceQueryInterval

The frequency at which IGMP Host-Query packets are transmitted on this interface.

J.2.1.2.1 Passive Mode

HFC-side: n/a, read-only. The CM MUST not transmit queries upstream. Return a value of zero.

CMCI-side: read only . This value is derived based on the interval of queries received from an upstream querier.

J.2.1.2.2 Active Mode

HFC-side: n/a, read-only. The CM MUST not transmit queries upstream. Return a value of zero.

CMCI-side: read-create. Min = 0; Max = $(2^{\wedge 32}-1)$; Default = 125

J.2.1.3 igmpInterfaceStatus

The activation of a row enables IGMP on the interface. The destruction of a row disables IGMP on the interface.

J.2.1.3.1 All Modes

MUST be enabled on both interfaces for all DOCSIS 1.1 CM interfaces.

J.2.1.4 igmpInterfaceVersion

The version of IGMP which is running on this interface. MUST be version 2 for all DOCSIS 1.1 CM interfaces.

J.2.1.5 igmpInterfaceQuerier

The address of the IGMP Querier on the IP subnet to which this interface is attached.

J.2.1.5.1 Passive Mode

HFC-side: read-only. MUST be the address of an upstream IGMP Querier device for both active and passive CMs.

CMCI-side: read-only. Same as HFC-side value.

J.2.1.5.2 Active Mode

HFC-side: read-only. MUST be the address of an upstream IGMP Querier device for both active and passive CMs.

CMCI-side: read-only. Active CMs may report it as the HFC-side value. However, active CM's that participate in IGMP Querier negotiation on the CMCI may report it as a different CPE.

J.2.1.6 *igmpInterfaceQueryMaxResponseTime*

The maximum query response time advertised in IGMPv2 queries on this interface.

J.2.1.6.1 Passive Mode

HFC-side: n/a, read-only. return a value of zero.

CMCI-side: read-only. This value is derived from observation of queries received from an upstream querier

J.2.1.6.2 Active Mode

HFC-side: n/a, read-only. return a value of zero.

CMCI-side: read-create. Min = 0; Max = 255; Default = 100.

J.2.1.7 igmpInterfaceQuerierUpTime

The time since igmpInterfaceQuerier was last changed.

J.2.1.7.1 PassiveMode

HFC-side: read-only.

CMC-side: n/a, read-only. Return a value of zero.

J.2.1.7.2 Active Mode

HFC-side: read-only.

CMCI-side: read-only.

J.2.1.8 igmpInterfaceQuerierExpiryTime

The amount of time remaining before the other querier present timer expires. If the local system is the querier, the value of this object is zero.

J.2.1.8.1 Passive Mode

Both interfaces: n/a, read-only. The CM is never the querier, return 0.

J.2.1.8.2 Active Mode

HFC-side: n/a, read-only. Return 0.

CMCI-side: read-only. The CM may only be the querier on the CMCI.

J.2.1.9 igmpInterfaceVersion1QuerierTimer

The time remaining until the host assumes that there are no IGMPv1 routers present on the interface. While this is non-zero, the host will reply to all queries with version 1 membership reports.

J.2.1.9.1 Passive Mode

HFC-side: n/a read-only. Return a value of zero.

CMCI-side: n/a read-only. Return a value of zero.

J.2.1.9.2 Active Mode

HFC-side: read-only.

CMCI-side: read-only.

J.2.1.10 igmpInterfaceWrongVersionQueries

The number of queries received whose IGMP version does not match igmpInterfaceVersion, over the lifetime of the row entry. IGMP requires that all routers on a LAN be configured to run the same version of IGMP. Although, DOCSIS 1.1 requires that all CM and CMTS devices support IGMPv2, it is possible for an upstream querier to be an IGMPv1 querier.

J.2.1.10.1 All Modes

All interfaces: read-only. The number of non-v2 queries received on this interface.

J.2.1.11 igmpInterfaceJoins

The number of times a group membership has been added on this interface; that is, the number of times an entry for this interface has been added to the Cache Table. This object gives an indication of the amount of IGMP activity over the lifetime of the row entry.

All HFC-side: n/a, read-only. Always return a value of zero (see CMCI-side).

IAll CMCI-side: read-only. Group membership is defined to only exist on the CMCI.

J.2.1.12 igmpInterfaceProxyIfIndex

Some devices implement a form of IGMP proxying whereby memberships learned on the interface represented by this row, cause IGMP Host Membership Reports to be sent on the interface whose ifIndex value is given by this object. Such a device would implement the igmpV2RouterMIBGroup only on its router interfaces (those interfaces with non-zero igmpInterfaceProxyIfIndex). Typically, the value of this object is 0, indicating that no proxying is being done.

J.2.1.12.1 Passive Mode

All Interfaces: read-only. Always return a value of zero.

J.2.1.12.2 Active Mode

HFC-side: read-only. Always return a value of zero.

CMCI-side: read-only. Always return a ifIndex for HFC-side interface.

J.2.1.13 igmpInterfaceGroups

The current number of entries for this interface in the Cache Table.

J.2.1.13.1 All HFC-side: n/a, read-only. Always return a value of zero (see CMCI-side).

J.2.1.13.2 All CMCI-side: read-only. Group membership is defined to only exist on the CMCI.

Number of active sessions Proxied or Active on this Interface.

J.2.1.14 igmpInterfaceRobustness

The robustness variable allows tuning for the expected packet loss on a subnet. If a subnet is expected to be lossy, the robustness variable may be increased. IGMP is robust to (robustness variable – 1) packet losses.

J.2.1.14.1 Passive Mode

HFC-side: n/a read-only. Return a value of zero.

CMCI-side: n/a read-only. Return a value of zero.

J.2.1.14.2 Active Mode

All interfaces: read-create. Min = 1; Max = (2^{32}-1); Default = 2

J.2.1.15 igmpInterfaceLastMemberQueryIntvl

The last member query interval is the max response time inserted into group specific queries sent in response to leave group messages, and is also the amount of time between group specific query messages. This value may be tuned to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group.

J.2.1.15.1 Passive Mode

HFC-side: n/a, read-only. return a value of zero.

CMCI-side: read-only. This value is derived from observation of queries received from an upstream querier

J.2.1.15.2 Active Mode

HFC-side: n/a, read-only. return a value of zero.

CMCI-side: read-create. Min = 0; Max = 255; Default = 100.

J.2.2 igmpCacheTable - igmpCacheEntry

J.2.2.1 igmpCacheAddress

The IP multicast group address for which this entry contains information.

J.2.2.1.1 All Modes

Not-accessible (index). Report the address of active IP Multicast on the CMCI interface.

J.2.2.2 igmpCacheIfIndex

The interface for which this entry contains information for an IP multicast group address.

J.2.2.2.1 All Modes

MUST only apply to CMCI interface (e.g., membership is only active on subscriber side of CM).

J.2.2.3 igmpCacheSelf

An indication of whether the local system is a member of this group address on this interface.

J.2.2.3.1 Passive Mode

read-only. MUST be set to FALSE. The CM is not a member of any group.

J.2.2.3.2 Active Mode

read-create. Implementation specific. If the CM is configured to be a member of the group, then membership reports are sent with the CM's IP Address but MUST ONLY be sent in proxy for active sessions on the CMCI (e.g., the CM MUST NOT be a member of a multicast group that is not active on the CMCI). If the CM is not configured to be a member, then the source IP Address of membership reports MUST be set to the current value of the igmpCacheLastReporter address.

J.2.2.4 igmpCacheLastReporter

The IP address of the source of the last membership report received for this IP Multicast group address on this interface. If no membership report has been received, this object has the value of 0.0.0.0.

J.2.2.4.1 All Modes

MUST only apply to last reporter on CMCI interface (e.g., membership is only active on subscriber side of CM).

J.2.2.5 igmpCacheUpTime

The time elapsed since this entry was created.

J.2.2.5.1 All Modes

read-only. MUST only apply to duration of membership on CMCI interface (e.g., membership is only active on subscriber side of CM).

J.2.2.6 igmpCacheExpiryTime

The minimum amount of time remaining before this entry will be aged out.

J.2.2.6.1 All Modes

read-only. MUST only apply to duration of membership on CMCI interface (e.g., membership is only active on subscriber side of CM).

J.2.2.7 igmpCacheStatus

The status of this entry.

J.2.2.7.1 All Modes

read-create. MUST only apply to membership on CMCI interface (e.g., membership is only active on subscriber side of CM). Deletion of a row results in preventing downstream forwarding to this IP Multicast group address on this interface.

J.2.2.8 igmpCacheVersion1HostTimer

The time remaining until the local querier will assume that there are no longer any IGMP version 1 members on the IP subnet attached to this interface. Upon hearing any IGMPv1 membership report, this value is reset to the group membership timer. While this time remaining is non-zero, the local querier ignores any IGMPv2 leave messages for this group that it receives on this interface.

J.2.2.8.1 Passive Mode

All interfaces: n/a, read-only. Return a value of zero.

J.2.2.8.2 Active Mode

HFC-side: n/a, read-only. Return a value of zero.

CMCI-side: read-only.

J.3 DOCSIS 1.1 CMTS Support for the IGMP MIB

There are two types of interfaces applicable to IGMP on the DOCSIS 1.1 CMTS. These are the NSI-Side and HFC-Side interfaces. Application of the IGMP MIB to DOCSIS 1.1 CMTSs is presented in terms of *passive* and *active* CMTS operation and these two interface types.

It is important to note that an *active* IGMP capable CMTS may be implemented as a proxy, router, or hybrid device. As such, the CMTS may be capable of querying on both its NSI and HFC side interfaces and may

manage membership for devices on its NSI interfaces (e.g., as a multicast router). This is different than an *active* CM, which MUST NOT query on its HFC side interface (e.g., it may only query on its CMCI). This capability is accounted for in the application of the IGMP MIB to the CMTS.

J.3.1 igmpInterfaceTable- igmpInterfaceEntry

J.3.1.1 *igmpInterfaceIfIndex*

The ifIndex value of the interface for which IGMP is enabled.

J.3.1.1.1 All Modes

This is the same for passive and active modes.

NSI-side: not-accessible. ifIndex of applicable network side interface(s).

HFC-side: not-accessible. ifIndex of docsCableMaclayer(127), CATV MAC Layer interface.

J.3.1.2 igmpInterfaceQueryInterval

The frequency at which IGMP Host-Query packets are transmitted on this interface.

J.3.1.2.1 Passive Mode

NSI-side: n/a, read-only. Return a value of zero.

HFC-side: read only . This value is derived based on the interval of queries received from a Network Side querier.

J.3.1.2.2 Active Mode

NSI-side: read-create. Min = 0; Max = $(2^{\wedge 32}-1)$; Default = 125

HFC-side: read-create. Min = 0; Max = $(2^{\wedge 32}-1)$; Default = 125

J.3.1.3 igmpInterfaceStatus

J.3.1.3.1 All Modes

The activation of a row enables IGMP on the interface. The destruction of a row disables IGMP on the interface.

J.3.1.4 igmpInterfaceVersion

The version of IGMP which is running on this interface. MUST be version 2 for all DOCSIS 1.1 CMTS interfaces.

J.3.1.5 igmpInterfaceQuerier

The address of the IGMP Querier on the IP subnet to which this interface is attached.

J.3.1.5.1 Passive Mode

NSI-side: read-only. This is the address of a network side device.

HFC-side: read-only. Same as NSI-side value.

J.3.1.5.2 Active Mode

NSI-side: read-only.

HFC-side: read-only. Active CMTSs MUST report this as an IP Address assigned to the CMTS' HFC-side interface. That is, queries MUST not originate from CMs or CPE.

J.3.1.6 igmpInterfaceQueryMaxResponseTime

The maximum query response time advertised in IGMPv2 queries on this interface.

J.3.1.6.1 Passive Mode

NSI-side: n/a, read-only. return a value of zero.

HFC-side: read-only. This value is derived from observation of queries received from a network side querier.

J.3.1.6.2 Active Mode

NSI-side: read-create. Min = 0; Max = 255; Default = 100.

HFC-side: read-create. Min = 0; Max = 255; Default = 100.

J.3.1.7 igmpInterfaceQuerierUpTime

The time since igmpInterfaceQuerier was last changed.

J.3.1.7.1 PassiveMode

NSI-side: read-only.

HFC-side: n/a, read-only. Return a value of zero.

J.3.1.7.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.8 igmpInterfaceQuerierExpiryTime

The amount of time remaining before the other querier present timer expires. If the local system is the querier, the value of this object is zero.

J.3.1.8.1 Passive Mode

Both interfaces: n/a, read-only. The CMTS is not the querier, return 0.

J.3.1.8.2 Active Mode

NSI-side: read-only.

HFC-side: read-only. The CMTS MUST be the only querier on the HFC.

J.3.1.9 igmpInterfaceVersion1QuerierTimer

The time remaining until the host assumes that there are no IGMPv1 routers present on the interface. While this is non-zero, the host will reply to all queries with version 1 membership reports.

J.3.1.9.1 Passive Mode

NSI-side: n/a read-only. Return a value of zero.

HFC-side: n/a read-only. Return a value of zero.

J.3.1.9.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.10 igmpInterfaceWrongVersionQueries

The number of queries received whose IGMP version does not match igmpInterfaceVersion, over the lifetime of the row entry. IGMP requires that all routers on a LAN be configured to run the same version of IGMP. Although, DOCSIS 1.1 requires that all CMTS and CMTSTS devices support IGMPv2, it is possible for a network side querier to be an IGMPv1 querier.

J.3.1.10.1 All Modes

All interfaces: read-only. The number of non-v2 queries received on this interface.

J.3.1.11 igmpInterfaceJoins

The number of times a group membership has been added on this interface; that is, the number of times an entry for this interface has been added to the Cache Table. This object gives an indication of the amount of IGMP activity over the lifetime of the row entry.

J.3.1.11.1 Passive Mode

NSI-side: n/a read-only. Return a value of zero.

HFC-side: n/a read-only. Return a value of zero.

J.3.1.11.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.12 igmpInterfaceProxyIfIndex

Some devices implement a form of IGMP proxying whereby memberships learned on the interface represented by this row, cause IGMP Host Membership Reports to be sent on the interface whose ifIndex value is given by this object. Such a device would implement the igmpV2RouterMIBGroup only on its router interfaces (those interfaces with non-zero igmpInterfaceProxyIfIndex). Typically, the value of this object is 0, indicating that no proxying is being done.

J.3.1.12.1 Passive Mode

All Interfaces: read-only. Always return a value of zero.

J.3.1.12.2 Active Mode

NSI-side: read-only.

HFC-side: read-only. Always return an ifIndex for a NSI-side interface.

J.3.1.13 igmpInterfaceGroups

The current number of entries for this interface in the Cache Table.

J.3.1.13.1 Passive Mode

NSI-side: n/a read-only. Return a value of zero.

HFC-side: n/a read-only. Group membership of HFC-side devices.

J.3.1.13.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.14 igmpInterfaceRobustness

The robustness variable allows tuning for the expected packet loss on a subnet. If a subnet is expected to be lossy, the robustness variable may be increased. IGMP is robust to (robustness variable – 1) packet losses.

J.3.1.14.1 Passive Mode

NSI-side: n/a read-only. Return a value of zero.

HFC-side: n/a read-only. Return a value of zero.

J.3.1.14.2 Active Mode

All interfaces: read-create. Min = 1; Max = ($2^{\wedge 32}$-1); Default = 2

J.3.1.15 igmpInterfaceLastMemberQueryIntvl

The last member query interval is the max response time inserted into group specific queries sent in response to leave group messages, and is also the amount of time between group specific query messages. This value may be tuned to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group.

J.3.1.15.1 Passive Mode

NSI-side: n/a, read-only. return a value of zero.

HFC-side: read-only. This value is derived from observation of queries received from a network side querier.

J.3.1.15.2 Active Mode

NSI-side: read-create. Min = 0; Max = 255; Default = 100.

HFC-side: read-create. Min = 0; Max = 255; Default = 100.

J.3.2 igmpCacheTable - igmpCacheEntry

J.3.2.1 igmpCacheAddress

The IP multicast group address for which this entry contains information.

J.3.2.1.1 All Modes

Not-accessible (index). Report the address of active IP Multicast on the interface.

J.3.2.2 igmpCacheIfIndex

The interface for which this entry contains information for an IP multicast group address.

J.3.2.2.1 Passive Mode

MUST only apply to HFC side interface (e.g., membership is only active on subscriber side of CMTS).

J.3.2.2.2 Active Mode

NSI-side: not-accessible

HFC-side: not-accessible

J.3.2.3 igmpCacheSelf

An indication of whether the local system is a member of this group address on this interface.

J.3.2.3.1 Passive Mode

read-only. MUST be set to FALSE. The CMTS is not a member of any group.

J.3.2.3.2 Active Mode

NSI-side: read-create. Implementation specific (i.e., may apply to RIPv2 or OSPF)

HFC-side: MUST be set to FALSE. The CMTS is not a member of any group on the HFC.

J.3.2.4 igmpCacheLastReporter

The IP address of the source of the last membership report received for this IP Multicast group address on this interface. If no membership report has been received, this object has the value of 0.0.0.0.

J.3.2.4.1 Passive Mode

MUST only apply to last reporter on HFC-side interface (e.g., membership is only active on subscriber side of CMTS).

J.3.2.4.2 Active Mode

NSI-side: read-only

HFC-side: read-only

J.3.2.5 igmpCacheUpTime

The time elapsed since this entry was created.

J.3.2.5.1 Passive Mode

MUST only apply to duration of membership on HFC-side interface (e.g., membership is only active on subscriber side of CMTS).

J.3.2.5.2 Active Mode

NSI-side: read-only

HFC-side: read-only

J.3.2.6 igmpCacheExpiryTime

The minimum amount of time remaining before this entry will be aged out.

J.3.2.6.1 Passive Mode

MUST only apply to duration of membership on HFC-side interface (e.g., membership is only active on subscriber side of CMTS).

J.3.2.6.2 Active Mode

NSI-side: read-only

HFC-side: read-only

J.3.2.7 igmpCacheStatus

The status of this entry.

J.3.2.7.1 Passive Mode

read-create MUST only apply to membership on HFC-side interface (e.g., membership is only active on subscriber side of CMTS). Deletion of a row results in preventing downstream forwarding to this IP Multicast group address on this interface.

J.3.2.7.2 Active Mode

NSI-side: read-create

HFC-side: read-create

J.3.2.8 igmpCacheVersion1HostTimer

The time remaining until the local querier will assume that there are no longer any IGMP version 1 members on the IP subnet attached to this interface. Upon hearing any IGMPv1 membership report, this value is reset to the group membership timer. While this time remaining is non-zero, the local querier ignores any IGMPv2 leave messages for this group that it receives on this interface.

J.3.2.8.1 Passive Mode

All interfaces: n/a, read-only. Return a value of zero.

J.3.2.8.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.3 IGMP MIB Compliance

J.3.3.1 docsIgmpV2PassiveDeviceCompliance

```
docsIgmpV2PassiveDeviceCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
    "The compliance statement for DOCSIS Devices passively running IGMPv2 and
     implementing the IGMP MIB."
    MODULE – this module
    MANDATORY-GROUPS { igmpBaseMIBGroup,
                       igmpRouterMIBGroup,
                       igmpV2RouterMIBGroup
                       }
    OBJECT igmpInterfaceStatus
    MIN-ACCESS read-only
    DESCRIPTION
    "Write access is not required."

    OBJECT igmpCacheStatus
    MIN-ACCESS read-only
    DESCRIPTION
    "Write access is not required."
```

```
    ::= {docsIgmpMIBCompliances 1}
```

J.3.3.2 docsIgmpV2ActiveDeviceCompliance

```
docsIgmpV2ActiveCmCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
    "The compliance statement for DOCSIS Devices actively running IGMPv2 and
     implementing the IGMP MIB."
    MODULE – this module
    MANDATORY-GROUPS { igmpBaseMIBGroup,
                       igmpV2HostMIBGroup,
                       igmpRouterMIBGroup,
                       igmpV2RouterMIBGroup
                       }
    OBJECT igmpInterfaceStatus
    MIN-ACCESS read-only
    DESCRIPTION
    "Write access is not required."

    OBJECT igmpCacheStatus
    MIN-ACCESS read-only
    DESCRIPTION
    "Write access is not required."

    ::= {docsIgmpMIBCompliances 2}
```

J.3.4 MIB Groups

See IGMP MIB for a description of the objects included in each group.

J.3.4.1 igmpV2HostMIBGroup

Active Devices only (optional – see notes for igmpCacheSelf).

J.3.4.2 igmpV2RouterMIBGroup

Active and Passive Devices

J.3.4.3 igmpBaseMIBGroup

Active and Passive Devices

J.3.4.4 igmpV2RouterMIBGroup

Active and Passive Devices

J.3.4.5 igmpRouterMIBGroup

Active and Passive Devices

J.3.4.6 igmpV2HostOptMIBGroup

Active and Passive Devices

J.3.4.7 igmpV2ProxyMIBGroup

Active Devices only.

Appendix K. Expected Behaviors for DOCSIS 1.1 modem in 1.0 and 1.1 modes in OSS area

The OSSI table Table 35 identifies DOCSIS OSSI 1.1 CM features that MAY and MUST be implemented in 1.0 mode.

Table 35. Expected Behaviors for DOCSIS 1.1 modem in 1.0 and 1.1 modes in OSS area

Specific requirement	Required behavior, DOCSIS 1.1 Modem in 1.0 Mode	Required behavior, DOCSIS 1.1 Modem in 1.1 Mode
Assignment of event-id	SHOULD support a 32-bit number with the following requirement: 1) Top bit is set to 0 for DOCSIS standard events; 2) top bit is set to 1 for vendor proprietary events.	MUST be a 32-bit number. Top bit is set to 0 for DOCSIS standard events. Top bit is set to 1 for vendor proprietary events.
Event Definitions	CM SHOULD support DOCSIS standard events defined in the OSSI 1.1 specification.	CM MUST support DOCSIS standard events defined in the OSSI 1.1 specification.
Default handling of events by priority. (Whether to store locally, send trap, or syslog message)	CM SHOULD behave as follow: Error and notice events are stored locally and sent as traps and syslog messages. Other event levels are stored only to the local log, except for informational and debug which are not stored or sent as traps or syslog messages.	CM MUST behave as follows: Error and notice events are stored locally and send traps and syslog messages. Other event levels store only to the local log, except for informational and debug which are not stored or cause any traps or syslog messages.
Meaning of event levels	CM SHOULD support event level definitions specified by the OSSI 1.1 specification.	CM MUST support event level definitions specified by the OSSI 1.1 specification.

Event storage in docsDevEventTable	Each entry in the dosDevEventTable contains an event-ID (identical to the Eventid requirement specified in section 4.4.2.2.2), event time stamp when the event occurred first time and last time, number of appearances and event description in human-readable English format. Total length of the each event description entry MUST not be longer then 255 characters (max. defined for SNMPadminString). Each event, or group of consecutive events with identical eventIds MUST constitute at least one row in the docsDevEvReporting table. For groups of consecutive events with identical eventIds, the CM MAY choose to store only a single row. In such a case, the event text of that row MUST match that of the most recent event. The event count MUST represent the number of events associated with that row. The first and last time columns MUST contain the time at which the least recent and most recent events associated with the row occurred respectively.	Each entry in the dosDevEventTable contains an event-ID (identical to the Eventid requirement specified in section 4.4.2.2.2), event time stamp when the event occurred first time and last time, number of appearances and event description in human-readable English format. Total length of the each event description entry MUST not be longer then 255 characters (max. defined for SNMPadminString). Each event, or group of consecutive events with identical eventIds MUST constitute at least one row in the docsDevEvReporting table. For groups of consecutive events with identical eventIds, the CM MAY choose to store only a single row. In such a case, the event text of that row MUST match that of the most recent event. The event count MUST represent the number of events associated with that row. The first and last time columns MUST contain the time at which the least recent and most recent events associated with the row occurred respectively.
Number of rows in docsDevEventTable	CM MUST support a minimum of 10 rows of docsDevEventTable.	CM MUST support a minimum of 10 rows of docsDevEventTable.
Event log persistence	Event log MUST persist across reboots	Event log MUST persist across reboots.
SNMP Version of Trap Control (when CM is in SNMP v1/v2c DocsDevNmAccess mode)	CM MUST implement docsDevNmAccessTrapVersion, which controls whether SNMP V1 or V2 traps are sent.	CM MUST implement docsDevNmAccessTrapVersion, which controls whether SNMP V1 or V2 traps are sent.
Syslog message format	CM SHOULD support the syslog message with the format: <level>CABLEMODEM [vendor]: <eventId> text OR <level>Cablemodem [vendor]: text	CM MUST support the syslog message with the format: <level>CABLEMODEM [vendor]: <eventId> text
SNMP Protocol Requirement	CM MUST support SNMP v1/v2c and SNMPv3 with DH. CM must support SNMP requirements specified in section 2.2 of the OSSI.	CM MUST support SNMP v1/v2c and SNMPv3 with DH
MIBs to implement	CM MUST support MIB objects as specified by Appendix A.	CM MUST support MIB objects as specified by Appendix A.

Deprecated MIB objects	Deprecated object is optional. If supported, the object MUST be implemented correctly. If not supported, the object MUST return appropriate SNMP error notifying that the object does not exist.	Deprecated object is optional. If supported, the object MUST be implemented correctly. If not supported, the object MUST return appropriate SNMP error notifying that the object does not exist.
Configuration Management	CM MUST support configuration management requirement as specified by Section 4.2 of the OSSI 1.1 specification.	CM MUST support configuration management requirement as specified by Section 4.2 of the OSSI 1.1 specification.
IP/LLC filters	CM SHOULD support LLC/IP filter requirement as specified by OSSI 1.1 specification.	CM MUST support LLC/IP filter requirement as specified by OSSI 1.1 specification.
CM interaction with CM configuration file	CM MUST process TLV type 11 entries in a configuration file as specified by Section 3.4 of the OSSI 1.1 specification.	CM MUST process TLV type 11 entries in a configuration file as specified by Section 3.4 of the OSSI 1.1 specification.
Additional MIB objects requirement	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 3.3 of the OSSI 1.1 specification.	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 3.3 of the OSSI 1.1 specification.
Performance management	CM MUST support performance management requirements as specified by Section 4.5 of the OSSI 1.1 specification.	CM MUST support performance management requirements as specified by Section 4.5 of the OSSI 1.1 specification.
OSS for CMCI	CM MUST support CMCI requirements as specified by Section 6 of the OSSI 1.1 specification.	CM MUST support CMCI requirements as specified by Section 6 of the OSSI 1.1 specification.

Appendix L. DOCS-IF-EXT-MIB

```
This MIB extends the RFC2670 DOCS-IF-MIB with three new objects defined.

The new object, docsIfDocsisCapability, is used to indicate the DOCSIS capability of a
cable device, that is whether it is DOCSIS1.1 capable or DOCsis1.0 capable.

The new object, docsIfDocsisOperMode, is used to indicate whether it is registered as a
DOCSIS1.1 device or DOCSIS1.0 device.

The new object, docsIfCmtsCmStatusDocsisMode, which augments the docsIfCmtsCmStatusTable
in DOCS-IF-MIB, is used to indicate whether a CM is registered as DOCSIS1.1 modem or
DOCSIS1.0 modem.

DOCS-IF-EXT-MIB DEFINITIONS ::= BEGIN

  IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE
                FROM SNMPv2-SMI
        OBJECT-GROUP,
        MODULE-COMPLIANCE
                FROM SNMPv2-CONF
       TEXTUAL-CONVENTION
              FROM SNMPv2-TC
        docsIfMib,
        docsIfCmtsCmStatusEntry
                FROM DOCS-IF-MIB;

docsIfExtMib MODULE-IDENTITY
        LAST-UPDATED    "0011160000Z" -- November 16, 2000
        ORGANIZATION    "IETF IPCDN Working Group"
        CONTACT-INFO
            "  "
        DESCRIPTION
            "This is the extension Module to rfc2670 DOCS-IF-MIB."
        REVISION "0010080000Z"
        DESCRIPTION
            "Initial Version. "
        ::= { docsIfMib 21 }

-- Textual Conventions
DocsisVersion ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION     "Indicates the docsis version number."
    SYNTAX          INTEGER {
                        docsis10 (1),
                        docsis11 (2)
                    }

docsIfDocsisCapability OBJECT-TYPE
        SYNTAX      DocsisVersion
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "Indication of the DOCSIS capability of the device.

             "
        ::= { docsIfExtMib 1 }

docsIfDocsisOperMode OBJECT-TYPE
        SYNTAX      DocsisVersion
        MAX-ACCESS  read-only
        STATUS      current
```

```
        DESCRIPTION
             "Indication whether the device has registered as a 1.0 or 1.1.

              For CMTS and unregistered CM, it is always the same as
              docsDevDocsisCapability.

             "
        ::= { docsIfExtMib 2 }


--
-- CM status table (within CMTS).
-- This table is implemented only at the CMTS.
-- It contains per CM status information available in the CMTS.
--

docsIfCmtsCmStatusExtTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF DocsIfCmtsCmStatusExtEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A set of objects in the CMTS, maintained for each
             Cable Modem connected to this CMTS."
        ::= { docsIfExtMib 3 }

docsIfCmtsCmStatusExtEntry OBJECT-TYPE
        SYNTAX      DocsIfCmtsCmStatusExtEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Status information for a single Cable Modem.
             An entry in this table exists for each Cable Modem
             which is connected to the CMTS."
        AUGMENTS { docsIfCmtsCmStatusEntry }
        ::= { docsIfCmtsCmStatusExtTable 1 }

DocsIfCmtsCmStatusExtEntry ::= SEQUENCE {
            docsIfCmtsCmStatusDocsisMode       DocsisVersion
        }

docsIfCmtsCmStatusDocsisMode OBJECT-TYPE
        SYNTAX      DocsisVersion
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indication whether the CM has registered as a 1.0 or 1.1 modem
"
        ::= { docsIfCmtsCmStatusExtEntry 1 }

docsIfExtConformance  OBJECT IDENTIFIER     ::= { docsIfExtMib 4 }
docsIfExtCompliances  OBJECT IDENTIFIER     ::= { docsIfExtConformance 1 }
docsIfExtGroups       OBJECT IDENTIFIER     ::= { docsIfExtConformance 2 }

-- compliance statements

docsIfExtCmCompliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION
            "The compliance statement."

MODULE  -- docsIfExtMib

-- unconditionally mandatory groups for CM
MANDATORY-GROUPS {
```

```
 docsIfDocsisVersionGroup
        }
        ::= { docsIfExtCompliances 1 }


docsIfDocsisVersionGroup OBJECT-GROUP
        OBJECTS {
             docsIfDocsisCapability,
             docsIfDocsisOperMode
        }
        STATUS      current
        DESCRIPTION
            "Object group to indicates DOCSIS version."
        ::= { docsIfExtGroups 1 }

docsIfExtCmtsCompliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION
            "The compliance statement."

MODULE  -- docsIfExtMib

-- unconditionally mandatory groups for CMTS

MANDATORY-GROUPS {
        docsIfExtGroup,
        docsIfDocsisVersionGroup
        }
        ::= { docsIfExtCompliances 2 }
docsIfExtGroup OBJECT-GROUP
        OBJECTS {
            docsIfCmtsCmStatusDocsisMode
        }
        STATUS      current
        DESCRIPTION
            "Mandatory implementation group for CMTS."
        ::= { docsIfExtGroups 2 }
```

END

Appendix M. DOCS-CABLE-DEVICE-TRAP-MIB

```
DOCS-CABLE-DEVICE-TRAP-MIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY,
NOTIFICATION-TYPE
FROM SNMPv2-SMI

MODULE-COMPLIANCE,
NOTIFICATION-GROUP
FROM SNMPv2-CONF

docsDev,
--docsDevBase,
docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsDevSwFilename,
docsDevSwServer,
docsDevServerDhcp,
docsDevServerTime,
docsDevNotification
FROM DOCS-CABLE-DEVICE-MIB --RFC2669

docsIfCmCmtsAddress,
docsIfCmtsCmStatusMacAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType,
docsIfCmtsCmStatusDocsisRegMode,
docsIfCmtsCmStatusModulationType
FROM DOCS-IF-MIB -- draft-ietf-ipcdn-docs-rfmibv2-05[84]

docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
docsIfCmtsCmStatusDocsisMode -- deprecated
FROM DOCS-IF-EXT-MIB -- deprecated

ifPhysAddress
FROM IF-MIB;
```

[84] Revised “02” to “05” per OSS-N-03022 by GO on 03/20/03.

docsDevTrapMIB MODULE-IDENTITY

LAST-UPDATED "0202250000Z"
ORGANIZATION "Cisco Systems, Inc."
CONTACT-INFO "
Junming Gao
Cisco Systems Inc
<jgao@ cisco. com>
"

DESCRIPTION
"Modified by David Raftus (david.raftus@imedia.com) to deprecate trap definition objects originating from the docsIfExt MIB. Corresponding objects from the Docsis 2.0 RF MIB draft were added to the trap definitions."

REVISION "000926000000Z"
DESCRIPTION
"The CABLE DEVICE TRAP MIB is an extension of the
CABLE DEVICE MIB defined in RFC2669.
It defines various trap objects for both cable
modem and cable modem termination systems.
Two groups of SNMP notification objects are defined.
One group is for notifying cable modem events and one group
for notifying cable modem termination system events.
Common to all CM notification objects (traps) is that
their OBJECTS statements contain information
about the event priority, the event Id, the event message
body, the CM DOCSIS capability, the CM DOCSIS QOS level, the CM DOCSIS upstream modulation type, the cable interface MAC address of the cable modem and the cable card MAC address of the CMTS to which the modem is connectede.

These objects are docsDevEvLevel, docsDevId, docsDevEvText,
docsIfDocsisBaseCapability, docsIfCmStatusDocsisOperMode, docsIfCmStatusModulationType,ifPhysAddress and docsIfCmCmtsAddress. The values of docsDevEvLevel, docsDevId, and docsDevEvText are from the entry which logs this event in the
docsDevEventTable, which is defined in
DOCS-CABLE-DEVICE-MIB of RFC2669. The docsIfDocsisBaseCapability, docsIfCmStatusDocsisOperMode, and docsIfCmStatusModulationType are defined in the DOCS-IF-MIB.
The ifPhysAddress value is the MAC address of the cable
interface of this cable modem. The docsIfCmCmtsAddress
specifies the MAC address of the CMTS (if there is a cable
card/ interface in the CMTS, then it is actually the
cable interface interface MAC address to which the CM is connected).
Individual CM trap may contain additional objects to

provide necessary information.

Common to all CMTS notification objects (traps) is that their OBJECTS statements contain information about the event priority, the event Id, the event message body, the connected CM DOCSIS QOS status, the connected CM DOCSIS modulation type, the CM cable interface MAC address, the CMTS DOCSIS capability, and the CMTS MAC address.

These objects are docsDevEvLevel, docsDevId, docsDevEvText, docsIfCmtsCmStatusDocsisRegMode, docsIfCmtsCmStatusModulationType, docsIfCmtsCmStatusMacAddress, docsIfDocsisBaseCapability, and ifPhysAddress. The values of docsDevEvLevel, docsDevId, and docsDevEvText are similar to those in CM traps. The values of docsIfCmtsCmStatusDocsisRegMode, docsIfCmtsCmStatusModulationType, and docsIfCmtsCmStatusMacAddress are from the docsIfCmtsCmStatusEntry (defined in DOCS-IF-MIB) corresponding to a connected CM. The docsIfDocsisBaseCapability indicates the CMTS DOCSIS capability.
The ifPhysAddress value is the CMTS MAC address (if there is a cable card/ interface in the CMTS, then it is actually the MAC address of the cable interface which connected to the CM).
"

::= { docsDev 10 }

--
--docsDevNotification OBJECT IDENTIFIER ::= { docsDev 2 }
--
docsDevTraps OBJECT IDENTIFIER ::= { docsDevNotification 1 }
docsDevTrapControl OBJECT IDENTIFIER ::= { docsDevTraps 1}
docsDevCmTraps OBJECT IDENTIFIER ::= { docsDevTraps 2 0 }
docsDevCmtsTraps OBJECT IDENTIFIER ::= { docsDevTraps 3 0 }

docsDevCmTrapControl OBJECT-TYPE

SYNTAX BITS {
cmInitTLVUnknownTrap(0),
cmDynServReqFailTrap(1),
cmDynServRspFailTrap(2),
cmDynServAckFailTrap(3),
cmBpiInitTrap(4),
cmBPKMTrap(5),
cmDynamicSATrap(6),

```
cmDHCPFailTrap( 7),
cmSwUpgradeInitTrap( 8),
cmSwUpgradeFailTrap( 9),
cmSwUpgradeSuccessTrap( 10),
cmSwUpgradeCVCTrap( 11),
cmTODFailTrap( 12),
cmDCCReqFailTrap( 13),
cmDCCRspFailTrap( 14),
cmDCCAckFailTrap( 15)
}

MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The object is used to enable CM traps. From left to right,
the set bit indicates the corresponding CM trap is enabled.
For example, if the first bit is set, then
docsDevCmInitTLVUnknownTrap is enabled. If it is zero,
the trap is disabled.
"

DEFVAL { '00'h }
::= { docsDevTrapControl 1 }


docsDevCmtsTrapControl OBJECT-TYPE

SYNTAX BITS {
cmtsInitRegReqFailTrap( 0),
cmtsInitRegRspFailTrap( 1),
cmtsInitRegAckFailTrap( 2),
cmtsDynServReqFailTrap( 3),
cmtsDynServRspFailTrap( 4),
cmtsDynServAckFailTrap( 5),
cmtsBpiInitTrap( 6),
cmtsBPKMTrap( 7),
cmtsDynamicSATrap( 8),
cmtsDCCReqFailTrap( 9),
cmtsDCCRspFailTrap( 10),
cmtsDCCAckFailTrap( 11)
}

MAX-ACCESS read-write
STATUS current
```

DESCRIPTION
"The object is used to enable CMTS traps. From left to right,
the set bit indicates the corresponding CMTS trap is enabled.
For example, if the first bit is set, then
docsDevCmtsInitRegRspFailTrap is enabled. If it is zero,
the trap is disabled.
"
DEFVAL { '00'h }
::= { docsDevTrapControl 2 }

docsDevCmInitTLVUnknownTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"Event due to detection of unknown TLV during
the TLV parsing process.
The values of docsDevEvLevel, docsDevId, and
docsDevEvText are from the entry which logs this event
in the docsDevEventTable. The docsIfDocsisBaseCapability
indicates the DOCSIS version information. The docsIfCmStatusDocsisOperMode indicates the QOS level of the CM, while the docsIfCmStatusModulationType indicates the upstream modulation methodology used by the CM.
The ifPhysAddress value is the MAC address of the cable interface of this cable modem.
The docsIfCmCmtsAddress specifies the MAC address
of the CMTS to which the CM is connected (if there is a cable
card/ interface in the CMTS, then it is actually the MAC address of the cable
interface which connected to the CM).
This part of information is uniformed across all CM traps.
"

::= { docsDevCmTraps 1 }

```
docsDevCmDynServReqFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability,  -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a dynamic service
request happened during the dynamic services process.
"

::= { docsDevCmTraps 2 }


docsDevCmDynServRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a dynamic service
response happened during the dynamic services process.
"

::= { docsDevCmTraps 3}


docsDevCmDynServAckFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
```

```
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability,  -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a dynamic service
acknowledgement happened during the dynamic services process.
"

::= { docsDevCmTraps 4}

docsDevCmBpiInitTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a BPI initialization
attempt happened during the registration process.
"

::= { docsDevCmTraps 5 }

docsDevCmBPKMTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
```

```
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a BPKM operation.
"

::= { docsDevCmTraps 6 }

docsDevCmDynamicSATrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a dynamic security
association operation.
"

::= { docsDevCmTraps 7 }

docsDevCmDHCPFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
```

```
docsDevEvText,
docsIfDocsisCapability,  -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevServerDhcp,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a DHCP server.
The value of docsDevServerDhcp is the IP address
of the DHCP server.
"

::= { docsDevCmTraps 8 }


docsDevCmSwUpgradeInitTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevSwFilename,
docsDevSwServer,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report a software upgrade initiated
event. The values of docsDevSwFilename, and
docsDevSwServer indicate the software image name
and the server IP address the image is from.
"
```

```
::= { docsDevCmTraps 9 }


docsDevCmSwUpgradeFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevSwFilename,
docsDevSwServer,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a software upgrade
attempt. The values of docsDevSwFilename, and
docsDevSwServer indicate the software image name
and the server IP address the image is from.
"

::= { docsDevCmTraps 10 }


docsDevCmSwUpgradeSuccessTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability,  -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevSwFilename,
docsDevSwServer,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }
```

```
STATUS current
DESCRIPTION
"An event to report the Software upgrade success event.
The values of docsDevSwFilename, and
docsDevSwServer indicate the software image name
and the server IP address the image is from.
"

::= { docsDevCmTraps 11 }


docsDevCmSwUpgradeCVCFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of the verification
of code file happened during a secure software upgrade
attempt.
"

::= { docsDevCmTraps 12 }


docsDevCmTODFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevServerTime,
docsIfDocsisBaseCapability,
```

```
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a time of day server.
The value of docsDevServerTime indicates the server IP
address.
"

::= { docsDevCmTraps 13 }


docsDevCmDCCReqFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic channel
change request happened during the dynamic channel
change process in the CM side.
"

::= { docsDevCmTraps 14 }


docsDevCmDCCRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
```

```
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic channel
change response happened during the dynamic channel
change process in the CM side.
"

::= { docsDevCmTraps 15 }


docsDevCmDCCAckFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability,  -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic channel
change acknowledgement happened during the dynamic channel
change process in the CM side.
"

::= { docsDevCmTraps 16}


docsDevCmtsInitRegReqFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
```

```
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current
DESCRIPTION
"An event to report the failure of a registration
request from CM happening during the CM initialization
process and detected on the CMTS side.
The values of docsDevEvLevel, docsDevId, and
docsDevEvText are from the entry which logs this event
in the docsDevEventTable. The docsIfCmtsCmStatusDocsisRegMode
and docsIfCmtsCmStatusMacAddress indicate the docsis
QOS version and the MAC address of the requesting CM. The docsIfCmtsCmModulationType
indicates the upstream modulation methodology used by the connected CM.
The docsIfDocsisBaseCapability and ifPhysAddress
indicate the docsis version of the CMTS and the MAC
address of the CMTS (if there is a cable
card/ interface in the CMTS, then it is actually the MAC address of the cable interface which
connected to the CM) cable card connected to the CM.
This part of information is uniformed across all CMTS traps.
"

::= { docsDevCmtsTraps 1 }


docsDevCmtsInitRegRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability,  -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
```

```
"An event to report the failure of a registration
response happened during the CM initialization
process and detected in the CMTS side.
"

::= { docsDevCmtsTraps 2 }


docsDevCmtsInitRegAckFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability,  -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a registration
acknowledgement from CM happened during the CM
initialization process and detected in the CMTS side.
"

::= { docsDevCmtsTraps 3 }


docsDevCmtsDynServReqFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability,  -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }
```

```
STATUS current
DESCRIPTION
"An event to report the failure of a dynamic service
request happened during the dynamic services process
and detected in the CMTS side.
"

::= { docsDevCmtsTraps 4 }


docsDevCmtsDynServRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic service
response happened during the dynamic services process
and detected in the CMTS side.
"

::= { docsDevCmtsTraps 5 }


docsDevCmtsDynServAckFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability,  -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }
```

```
STATUS current
DESCRIPTION
"An event to report the failure of a dynamic service
acknowledgement happened during the dynamic services
process and detected in the CMTS side.
"

::= { docsDevCmtsTraps 6 }


docsDevCmtsBpiInitTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability,  -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a BPI initialization
attempt happened during the CM registration process
and detected in the CMTS side.
"

::= { docsDevCmtsTraps 7 }


docsDevCmtsBPKMTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
```

```
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a BPKM operation
which is detected in the CMTS side.
"

::= { docsDevCmtsTraps 8 }


docsDevCmtsDynamicSATrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic security
association operation which is detected in the CMTS side.
"

::= { docsDevCmtsTraps 9 }


docsDevCmtsDCCReqFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
```

```
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic channel
change request happened during the dynamic channel
change process in the CM side and detected in the
CMTS side.
"

::= { docsDevCmtsTraps 10 }


docsDevCmtsDCCRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic channel
change response happened during the dynamic channel
change process in the CMTS side.
"

::= { docsDevCmtsTraps 11 }


docsDevCmtsDCCAckFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
```

```
docsIfDocsisCapability,  -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }


STATUS current
DESCRIPTION
"An event to report the failure of a dynamic channel
change acknowledgement happened during the dynamic channel
change process in the CMTS side.
"

::= { docsDevCmtsTraps 12}
--
--Conformance definitions
--
docsDevTrapConformance OBJECT IDENTIFIER ::= { docsDevTraps 4 }
docsDevTrapGroups OBJECT IDENTIFIER ::= { docsDevTrapConformance 1 }
docsDevTrapCompliances OBJECT IDENTIFIER ::= { docsDevTrapConformance 2 }
docsDevCmTrapCompliance MODULE-COMPLIANCE

STATUS current
DESCRIPTION
"The compliance statement for Cable Modem Traps and Control"

MODULE --docsDevTrap
--mandatory groups

GROUP docsDevCmTrapControlGroup
DESCRIPTION
"Mandatory in CM."

GROUP docsDevCmNotificationGroup
DESCRIPTION
"Mandatory in Cable Modem."

::= { docsDevTrapCompliances 1 }

docsDevCmTrapControlGroup OBJECT-GROUP
OBJECTS {
docsDevCmTrapControl
}
```

```
STATUS current
DESCRIPTION
"CM must support docsDevCmTrapControl."
::= { docsDevTrapGroups 1 }

docsDevCmNotificationGroup NOTIFICATION-GROUP

NOTIFICATIONS {
docsDevCmInitTLVUnknownTrap,
docsDevCmDynServReqFailTrap,
docsDevCmDynServRspFailTrap,
docsDevCmDynServAckFailTrap,
docsDevCmBpiInitTrap,
docsDevCmBPKMTrap,
docsDevCmDynamicSATrap,
docsDevCmDHCPFailTrap,
docsDevCmSwUpgradeInitTrap,
docsDevCmSwUpgradeFailTrap,
docsDevCmSwUpgradeSuccessTrap,
docsDevCmSwUpgradeCVCFailTrap,
docsDevCmTODFailTrap,
docsDevCmDCCReqFailTrap,
docsDevCmDCCRspFailTrap,
docsDevCmDCCAckFailTrap
}

STATUS current
DESCRIPTION
"A collection of CM notifications providing device status and
control."

::= { docsDevTrapGroups 2 }

docsDevCmtsTrapCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"The compliance statement for MCNS Cable Modems and
Cable Modem Termination Systems."
MODULE --docsDevTrap
--mandatory groups

GROUP docsDevCmtsTrapControlGroup
DESCRIPTION
"Mandatory in CMTS."
```

```
GROUP docsDevCmtsNotificationGroup
DESCRIPTION
"Mandatory in Cable Modem Termination Systems."

::= { docsDevTrapCompliances 2 }

docsDevCmtsTrapControlGroup OBJECT-GROUP
OBJECTS {
docsDevCmtsTrapControl
}

STATUS current
DESCRIPTION
"CMTS must support docsDevCmtsTrapControl."
::= { docsDevTrapGroups 3 }
docsDevCmtsNotificationGroup NOTIFICATION-GROUP

NOTIFICATIONS {
docsDevCmtsInitRegReqFailTrap,
docsDevCmtsInitRegRspFailTrap,
docsDevCmtsInitRegAckFailTrap ,
docsDevCmtsDynServReqFailTrap,
docsDevCmtsDynServRspFailTrap,
docsDevCmtsDynServAckFailTrap,
docsDevCmtsBpiInitTrap,
docsDevCmtsBPKMTrap,
docsDevCmtsDynamicSATrap,
docsDevCmtsDCCReqFailTrap,
docsDevCmtsDCCRspFailTrap,
docsDevCmtsDCCAckFailTrap
}

STATUS current
DESCRIPTION
"A collection of CMTS notifications providing device status and
control."
::= { docsDevTrapGroups 4 }
END
```

Appendix N. References[85]

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[ID-IGMP] Fenner, W., IGMP-based Multicast Forwarding (``IGMP Proxying''), IETF Internet Draft,. http://www.ietf.org/internet-drafts/draft-ietf-magma-igmp-proxy-00.txt

[IETF4] Mike Patrick, "Data Over Cable System Quality of Service Management Information Base", draft-ietf-ipcdn-qos-mib-04.txt, Oct 18, 2000, http://www.ipcdn.org/ipcdn-ids.html

[IETF6] Proposed Standard RFC version of BPI+ MIB, “draft-ietf-ipcdn-bpiplus-05.txt", http://www.ipcdn.org/ipcdn-ids.html

[IETF7] Proposed Standard RFC version of USB MIB, “dolnik-usb-mib-00.txt”

[IETF9] Proposed Standard RFC version of Customer Management MIB, “draft-ietf-ipcdn-subscriber-mib-02.txt”, http://www.ipcdn.org/ipcdn-ids.html

[IETF10] S. Cheshire and B. Aboba, “Dynamic Configuration of IPv4 Link-Local Addresses”, nternet-draft draft-ietf-zeroconf-ipv4-linklocal-05.txt.

[ITEF11] Aviv Goren/David Raftus; “Radio Frequency (RF) Interface Management Information Base for DOCSIS 2.0 compliant RF interfaces”; draft-ietf-ipcdn-docs-rfmibv2-05.txt.86, http://www.ipcdn.org/ipcdn-ids.html

[DOCSIS 1] DOCSIS Cable Modem Termination System - Network-Side Interface Specification SP-CMTS-NSI-I01-960702

[DOCSIS 2] DOCSIS Cable Modem to Customer Premise Equipment Interface Specification SP-CMCI-I10-050408

[DOCSIS 4] DOCSIS Data Over Cable Services Cable Modem Telephony Return Interface Specification SP-CMTRI-I01-970804

[DOCSIS 5] DOCSIS Data Over Cable Services Cable Modem Radio Frequency Interface Specification SP-RFIv1.1-C01-050907

[DOCSIS 6] DOCSIS Baseline Privacy Plus Interface Specification SP-BPI+-I12-050812[87]

[RFC-1157] Schoffstall, M., Fedor, M., Davin, J. and Case, J., A Simple Network Management Protocol (SNMP), IETF RFC-1157, May, 1990

[85] Appendix N updated per ECN OSS-N-02190 by GO on 11/15/02.

[86] Added this Reference per ECN OSS-N-03068 by GO on 07/11/03.

[87] Corrected spec number per ECN OSS-N-03020 by GO on 03/21/03.

[RFC-1213] K. McCloghrie and M. Rose. Management Information Base for Network Management of TCP/IP-base internets: MIB-II, IETF RFC-1213, March, 1991

[RFC-1224] L. Steinberg., Techniques for Managing Asynchronously Generated Alerts, IETF RFC-1224, May, 1991

[RFC-1493] E. Decker, P. Langille, A. Rijsinghani, and K.McCloghrie., Definitions of Managed Objects for Bridges, IETF RFC-1493, July, 1993

[RFC-1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC-1901, January 1996.

[RFC-3416] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC-3417] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for the Simple Network Management Protocol", STD 62, RFC 3417, December 2002.

[RFC-3418] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC-2011] K. McCloghrie, "Category: Standards Track SNMPv2 Management Information Base for the Internet Protocol using SMIv2", November 1996

[RFC-2013] K. McCloghrie, "Category: Standards Track SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2", November 1996

[RFC-2132] S. Alexander, R. Droms. DHCP Options and BOOTP Vendor Extensions. IETF RFC-2132. March, 1997.

[RFC-2863] K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB ", June 2000.

[RFC-2358] J. Flick, J. Johnson, "Definitions of Managed Objects for the Ethernet-like Interface Types", June 1998

[RFC-2570] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", April 1999

[RFC-2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC-2571, April 1999.

[RFC-2572] Case, J., Harrington, D., Presuhn, R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC-2572, April 1999

[RFC-2573] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC-2573, April 1999.

[RFC-2574] Blumenthal, U. and B. Wijnen, "The User-Based Security Model for Version 3 of the Simple Network Management Protocol (SNMPv3)", RFC-2574, April 1999

[RFC-2575] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model for the Simple Network Management Protocol (SNMP)", RFC-2575, April 1999

[RFC-2576] R. Frye, D. Levi, S. Routhier, B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard and Network Management Framework", RFC-2576, March 2000.

[RFC-2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC-2578, April 1999

[RFC-2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC-2579, April 1999

[RFC-2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC-2580, April 1999

[RFC-2669] M. St. Johns, "DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems", August 1999

[RFC-2670] M. St. Johns, "Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces", August 1999

[RFC-2786] M. St. Johns, "Diffie-Helman USM Key Management Information Base and Textual Convention", March, 2000

[RFC-2933] McCloghrie, K., Farinacci, D., Thaler, D., “Internet Group Management Protocol MIB”, RFC-2933

[RFC-3083] R. Woundy, “Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems”, RFC3083, March 2001.

Appendix O. Acknowledgements

The following contributors deserve genuine gratitude for their efforts in the development of the OSSI 1.1 specification.

Pak Siripunkaw of AT&T

Taft Singletary of Cox Communications

Jason Schnitzer of Stargus

Mike St. Johns of @Home

Asha Hegde of Cisco

Daniel Chuang of 3Com

Minnie Lu of Cisco

Bill Yost of TCE

Kazuyoshi Ozawa of Toshiba

Bob Himlin of TurboNet

Douglas Jones of YAS

Tasheer Syed of TCE

Benjamin Dolnik of 3Com

Randy Demuynck of Ericsson

Raymond Hou of Amplifynet

Fred Kiremidjian of Amplifynet

Adam Parmelee of Terayon

Dan Smith of ADC Telecommunications, Inc.

Pavaz Kokan of TCE

CableLabs and the cable industry as a whole are grateful to these individuals and organizations for their contributions. Their diligent work and professional approach should be commended and their continued enthusiasm will be invaluable as the OSSI specification evolves.

Appendix P. Revisions

R.1 ECNs included in SP-OSSIv1.1-I01-000407

ECN	Date Accepted	Author
oss-n-00011	03/01/00	Pak Siripunkaw
oss-n-00027	04/19/00	Kaz Ozawa

R.2 ECNs included in SP-OSSIv1.1-I02-000714

ECN	Date Accepted	Author
oss-n-00039	05/10/00	William Yost
oss-n-00040	05/17/00	William Yost
oss-n-00041	05/17/00	Minnie Lu
oss-n-00054	06/21/00	Pak Siripunkaw

R.3 ECNs included in SP-OSSIv1.1-I03-001215

ECN	Date Accepted	Author
oss-n-00063	07/26/00	Pak Siripunkaw
oss-n-00065	08/02/00	Dan Smith
oss-n-00066	08/02/00	Pak Siripunkaw
oss-n-00067	08/09/00	Pak Siripunkaw
oss-n-00068	08/16/00	Kaz Ozawa
oss-n-00074	09/13/00	Erich Arnold
oss-n-00077	10/11/00	William H. Yost
oss-n-00078	11/15/00	Pak Siripunkaw
oss-n-00080	10/04/00	Dan Smith
oss-n-00081	09/27/00	David Raftus
oss-n-00087	11/08/00	Pak Siripunkaw
oss-n-00090	10/25/00	Dan Smith
oss-n-00091	11/08/00	Erich Arnold
oss-n-00094	11/08/00	Dan Smith
oss-n-00096	11/22/00	Pak Siripunkaw
oss-n-00097	11/08/00	Pak Siripunkaw
oss-n-00102	11/15/00	Bruce Braidek
oss-n-00106	11/15/00	Greg Nakanishi
oss-n-00107	11/15/00	Dan Smith
oss-n-00109	11/22/00	David Raftus
oss-n-00110	11/22/00	Pak Siripunkaw
oss-n-00111	11/22/00	Pak Siripunkaw
oss-n-00117	11/22/00	Dan Smith

R.4 ECNs included in SP-OSSIv1.1-I04-010829

ECN	Date Accepted	Author
oss-n-00108	02/14/01	Junming Gao
oss-n-00118	12/27/00	David Raftus
oss-n-00129	01/03/01	David Raftus
oss-n-00130	01/24/01	Kaz Ozawa
oss-n-00134	01/17/01	Dan Smith
oss-n-00135	01/17/01	David Raftus
oss-n-01006	02/07/01	Dan Smith
oss-n-01007	02/07/01	Dan Smith
oss-n-01015	03/07/01	Pak Siripunkaw
oss-n-01016	03/21/01	Diego Mazzola
oss-n-01018	03/07/01	Kaz Ozawa
oss-n-01019	01/07/01	Lior Levy
oss-n-01020	03/14/01	Chris Thierman
oss-n-01023	03/28/01	Chris Thierman
oss-n-01024	03/28/01	Gordon Li
oss-n-01025	05/02/01	David Raftus
oss-n-01037	04/18/01	Kaz Ozawa
oss-n-01050	05/23/01	Kaz Ozawa
oss-n-01056	05/23/01	Kaz Ozawa
oss-n-01066	06/27/01	David Raftus
oss-n-01070	06/13/01	Kaz Ozawa

R.5 ECNs included in SP-OSSIv1.1-I05-020301

ECN	Date Accepted	Author
oss-n-01033	11/14/2001	Minnie Lu
oss-n-01052	11/7/2001	Lisa Ruby
oss-n-01067	11/7/2000	Margo Dolas
oss-n-01076	10/17/01	Dmitrii Loukianov
oss-n-01079	10/31/01	Azlina Ahmad
oss-n-01084	10/31/01	Lucy Pollak
oss-n-01089	10/31/01	David Raftus
oss-n-01097	11/21/01	Greg Nakanishi
oss-n-01099	11/28/01	André Lejeune
oss-n-01103	1/9/02	Dan Smith
oss-n-01104	1/3/02	Ofer Miranda

R.6 ECNs included in SP-OSSIv1.1-I06-020830

ECN	Date Accepted	Author
OSS-N-02009	2/27/02	Kaz Ozawa
OSS-N-02012	2/20/02	Eduardo Cardona
OSS-N-02017	2/27/02	David Raftus
OSS-N-02021	3/06/02	Anik Lacerte
OSS-N-02022	3/06/02	André Lejeune
OSS-N-02023	3/13/02	André Lejeune
OSS-N-02060	5/01/02	Lisa Ruby
OSS-N-02061	4/10/02	John Ulvr
OSS-N-02063	4/24/02	Lior Levy
OSS-N-02067	5/01/02	Kaz Ozawa
OSS-N-02068	5/01/02	Steven Cotton
OSS-N-02101	5/22/02	Harold Roberts
OSS-N-02106	5/22/02	David Raftus
OSS-N-02110	6/12/02	Anik Lacerte
OSS-N-02126	7/10/02	David Raftus
OSS-N-02146	8/14/02	Miron Tzhori
OSS-N-02148	8/14/02	Margo Dolas
OSS-N-02150	8/14/02	Eduardo Cardona
OSS-N-02153	8/14/02	John Ulvr
OSS-N-02155	8/14/02	Eduardo Cardona
OSS-N-02156	8/14/02	Eduardo Cardona
OSS-N-02157	8/14/02	Eduardo Cardona
OSS-N-02170	8/14/02	Greg White

R.7 ECNs included in SP-OSSIv1.1-I07-030730

ECN	Date Accepted	Author
oss-n-02136	8/28/02	Erich Arnold
oss-n-02141	8/21/02	Kirk Friedman
oss-n-02167	8/28/02	Eduardo Cardona
oss-n-02174	9/18/02	Greg Nakanishi
oss-n-02190	11/13/02	Eduardo Cardona
oss-n-02192	12/04/02	Eduardo Cardona
oss-n-02197	11/13/02	Erich Arnold
oss-n-02198	11/20/02	Matt Schmitt
oss-n-02204	11/6/02	Alexander Katsnelson
oss-n-02213	11/27/02	Eduardo Cardona
oss-n-02214	11/20/02	Minnie Lu
oss-n-02219	11/27/02	Minnie Lu
oss-n-02229	01/02/03	Alexander Katsnelson
oss-n-03005	03/05/03	Patrick Howard
oss-n-03006	04/09/03	Michael McFarland
oss-n-03009	02/12/03	Kevin A. Marez
oss-n-03013	02/19/03	Kevin A. Marez
oss-n-03020	03/12/03	Alexander Katsnelson
oss-n-03022	03/12/03	Lina Nakhle
oss-n-03024	6/11/03	Eduardo Cardona
oss-n-03045	04/30/03	Kevin A. Marez
oss-n-03054	05/28/03	Alexander Katsnelson
oss-n-03064	7/3/03	Lucy Pollak
oss-n-03066	7/2/03	Eduardo Cardona
oss-n-03068	7/2/03	Eduardo Cardona
oss-n-03070	7/2/03	Eduardo Cardona

R.8 ECNs included in CM-SP-OSSIv1.1-C01-050907

ECN	Date Accepted	Author
OSSIv1.1-N-05.0241-1	8/31/05	Greg White