Data-Over-Cable Service Interface Specifications
DOCSIS 2.0

Operations Support System Interface Specification

SP-OSSIv2.0-I05-040407

ISSUED
SPECIFICATION

Document Status Sheet

Document Control Number:	SP-OSSIv2.0-I05-040407			
Reference:	Operations Support System Interface Specification			
Revision History:	I01 — First Issued Release, December 31, 2001 I02 — Second Issued Release, June 17, 2002 I03 — Third Issued Release, December 18, 2002 I04 — Fourth Issued Release, July 30, 2003 I05 — Fifth Issued Release, April 7, 2004			
Date:	April 7, 2004			
Status Code:	Work in Process	Draft	Issued	Closed
Distribution Restrictions:	CableLabs only	CL Reviewers	CL Vendor	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 cable industry 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.

Table of Contents

This page intentionally left blank.

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

1 Scope and Purpose

1.1 Scope

This Specification defines the Network Management requirements to support a DOCSIS® 2.0 environment. More specifically, the specification details the SNMPv3 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 principles 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 word 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 or 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 weighed 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 requires 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.

This page intentionally left blank.

2 References (normative/informative)[1]

[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-I09-030730

[DOCSIS 4] DOCSIS Telephony Return Interface Specification SP-CMTRI-I01-970804

[DOCSIS 5] DOCSIS Radio Frequency Interface Specification SP-RFIv2.0-I04-030730

[DOCSIS 6] DOCSIS Baseline Privacy Plus Interface Specification SP-BPI+-I10-030730

[ID-IGMP] Fenner, W., IGMP-based Multicast Forwarding ("IGMP Proxying"), IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-fenner-igmp-proxy-03.txt.

[IETF3] draft-ietf-ipcdn-igmp-mib-00.txt, H. Abramson, "DOCSIS 1.1 IGMP MIB", June 1999

[IETF4] draft-ietf-ipcdn-qos-mib-04.txt, Mike Patrick, "Data Over Cable System Quality of Service Management Information Base", Oct. 18, 2000

[IETF6] Proposed Standard RFC version of BPI+ MIB, "draft-ietf-ipcdn-bpiplus-05.txt"[2]

[IETF7] Proposed Standard RFC version of USB MIB, "draft-ietf-xxxx-xxxx-xxxx-00.txt"

[IETF9] Proposed Standard RFC version of Customer Management MIB, "draft-ietf-ipcdn-subscriber-mib-02.txt"

[IETF10] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", internet-draft draft-ietf-zeroconf-ipv4-linklocal-10.txt[3]

[IETF11] Raftus David, Goren Aviv, "Proposed Standard RFC version of Radio Frequency (RF) Interface Mib", draft-ietf-ipcdn-docs-rfmibv2-05.txt[4]

[NDM-U 3.1] "Network Data Management – Usage (NDM-U) For IP-Based Services", Version 3.1, IPDR.org, April 15, 2002.

[RFC 1157] Schoffstall, M., Fedor, M., Davin, J. and Case, J., A Simple Network Management Protocol (SNMP), IETF RFC 1157, May, 1990

[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

---

1. Deleted reference to RFI-MIB-IPCDN-DRAFT per ECN OSS2-N-03069 by GO on 07/11/03.

2. Revised this reference (and rescinded ECN OSS2-N-02234) per ECN OSS2-N-03021 by GO on 03/21/03.

3. Revised this reference per ECN OSSIv2.0-N-03.0117-2 by GO on 02/19/04.

4. Added this reference per ECN OSS2-N-03069 by GO on 07/11/03.

[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 1952] Deutsch, P., "GZIP file format specification version 4.3", RFC 1952, May, 1996.

[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 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 2665] J. Flick, J. Johnson, "Definitions of Managed Objects for the Ethernet-like Interface Types", August 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] stjohns-snmpv3-dhkeychange-mib-01.txt, Michael C. St. Johns, "Diffie-Helman USM Key MIB August 1999 Diffie-Helman USM Key Management Information Base and Textual Convention", Aug. 6, 1999

[RFC 2863] K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB", June 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", RFC 3083, March 2001.

[RFC 3410] J. Case, R. Mundy, D. Partain, B. Stewart, Introduction and Applicability Statements for Internet-Standard Management Framework, December 2002.

[RFC 3411] D. Harrington, R. Presuhn, B. Wijnen, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, December 2002.

[RFC 3412] J. Case, D. Harrington, R. Presuhn, B. Wijnen, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3413] D. Levi, P. Meyer, B. Stewart, Simple Network Management Protocol (SNMP) Applications, December 2002.

[RFC 3414] U. Blumenthal, B. Wijnen, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), December 2002.

[RFC 3415] B. Wijnen, R. Presuhn, K. McCloghrie, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3416] R. Presuhn, Ed., Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3417] R. Presuhn, Ed., Transport Mappings for the Simple Network Management Protocol (SNMP), December 2002.

[RFC 3418] R. Presuhn, Ed., Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), December 2002.

[SYSLOG] R. C. Lonvick, "The BSD syslog Protocol", Informational RFC 3164, August 2001.[1]

1. Added reference per ECN OSSIv2.0-N-03.0117-2 by GO on 2/19/04.

This page intentionally left blank.

3 Glossary (informative)

Active Service Flow An admitted Service Flow from the CM to the CMTS which is available for packet transmission.

Address Resolution Protocol (ARP) A protocol of the IETF for converting network addresses to 48-bit Ethernet addresses.

Admitted Service Flow A Service Flow, either provisioned or dynamically signaled, which is authorized and for which resources have been reserved but is not active.

Allocation A group of contiguous mini-slots in a MAP which constitute a single transmit opportunity.

American National Standards Institute (ANSI) A US standards body.

Asynchronous Transfer Mode (ATM) A protocol for the transmission of a variety of digital signals using uniform 53-byte cells.

A-TDMA DOCSIS 2.0 TDMA mode (as distinguished from DOCSIS 1.x TDMA).

Authorization Module The authorization module is an abstract module that the CMTS can contact to authorize Service Flows and Classifiers. The authorization module tells the CMTS whether the requesting CM is authorized for the resources it is requesting.

Availability In cable television systems, availability is the long-term ratio of the actual RF channel operation time to scheduled RF channel operation time (expressed as a percent value) and is based on a bit error rate (BER) assumption.

Bandwidth Allocation Map The MAC Management Message that the CMTS uses to allocate transmission opportunities to CMs.

Bridge Protocol Data Unit (BDU) Spanning tree protocol messages as defined in [ISO/IEC10038].

Broadcast Addresses A predefined destination address that denotes the set of all data network service access points.

Burst A single continuous RF signal from the upstream transmitter, from transmitter on to transmitter off.

Burst Error Second Any Errored Second containing at least 100 errors.

Cable Modem (CM) A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.

Cable Modem Termination System (CMTS) Cable modem termination system, located at the cable television system head-end or distribution hub, which provides complementary functionality to the cable modems to enable data connectivity to a wide-area network.

Cable Modem Termination System - Network Side Interface (CMTS-NSI) The interface, defined in [DOCSIS 1], between a CMTS and the equipment on its network side.

Cable Modem to CPE Interface (CMCI) The interface, defined in [DOCSIS 4], between a CM and CPE.

Carrier Hum Modulation The peak-to-peak magnitude of the amplitude distortion relative to the RF carrier signal level due to the fundamental and low-order harmonics of the power-supply frequency.

Carrier-to-Noise Ratio (C/N or CNR) The ratio of signal power to noise power in the defined measurement bandwidth. For digital modulation, CNR = E_s/N_0, the energy-per-symbol to noise-density ratio; the signal power is measured in the occupied bandwidth, and the noise power is normalized to the modulation-rate bandwidth. For video, the measurement bandwidth is 4 MHz.

CCCM CPE Controlled Cable Modem. Refer to the DOCSIS Cable Modem to Customer Premise Equipment Interface (CMCI) specification.

Channel The frequency spectrum occupied by a signal. Usually specified by center frequency and bandwidth parameters.

Chip Each of the 128 bits comprising the S-CDMA spreading codes.

Chip Duration The time to transmit one chip of the S-CDMA spreading code. The inverse of the chip rate.

Chip Rate The rate at which individual chips of the S-CDMA spreading codes are transmitted. (1280 to 5120 kHz)

Classifier A set of criteria used for packet matching according to TCP, UDP, IP, LLC, and/or 802.1P/Q packet fields. A classifier maps each packet to a Service Flow. A Downstream Classifier is used by the CMTS to assign packets to downstream service flows. An Upstream Classifier is used by the CM to assign packets to upstream service flows.

Code Hopping Matrix A shifted version of the reference code matrix (see below) that is used when code hopping is employed to vary the codes used by each CM. The Code Hopping Matrix is either 128 rows by 128 columns (when all 128 codes are active) or is 127 rows by 128 columns (when less than 128 codes are active in the S-CDMA spreader-on frame). When less than 128 codes are active, Code 0 (all ones) is deleted from the matrix, but all remaining codes are still cycled through even if less than 127 codes are active in a frame.

Composite Second Order Beat (CSO) The peak of the average level of distortion products due to second-order non-linearities in cable system equipment.

Composite Triple Beat (CTB) The peak of the average level of distortion components due to third-order non-linearities in cable system equipment.

Cross-Modulation A form of television signal distortion where modulation from one or more television channels is imposed on another channel or channels.

Customer See End User.

Customer Premises Equipment (CPE) Equipment at the end user’s premises; MAY be provided by the end user or the service provider.

Data Link Layer Layer 2 in the Open System Interconnection (OSI) architecture; the layer that provides services to transfer data over the transmission link between open systems.

Distribution Hub A location in a cable television network which performs the functions of a head-end for customers in its immediate area, and which receives some or all of its television program material from a Master Head-end in the same metropolitan or regional area.

Downstream In cable television, the direction of transmission from the head-end to the subscriber.

Drop Cable Coaxial cable that connects to a residence or service location from a directional coupler (tap) on the nearest coaxial feeder cable.

Dynamic Host Configuration Protocol (DHCP) An Internet protocol used for assigning network-layer (IP) addresses.

Dynamic Range The ratio between the greatest signal power that can be transmitted over a multichannel analog transmission system without exceeding distortion or other performance limits, and the least signal power that can be utilized without exceeding noise, error rate or other performance limits.

Electronic Industries Association (EIA) A voluntary body of manufacturers which, among other activities, prepares and publishes standards.

End User A human being, organization, or telecommunications system that accesses the network in order to communicate via the services provided by the network.

Engineering Change Notice The final step in the procedure to change specifications.

Engineering Change Order The second step in the procedure to change specifications. DOCSIS posts ECO to web site EC table and ECO page (with indication of ECO Comment Deadline). DOCSIS issues ECO announcement to DOCSIS-announce and working group mail lists (with indication of ECO Comment Deadline).

Engineering Change Request The first step in the procedure to change specifications. DOCSIS issues ECR number, posts to web site EC table and ECR page. DOCSIS sends ECR to subject area working group mail list (and author).

Errored Second Any 1-sec interval containing at least one bit error.

Extended Subsplit A frequency division scheme that allows bidirectional traffic on a single coaxial cable. Reverse path signals come to the head-end from 5 to 42 MHz. Forward path signals go from the head-end from 50 or 54 MHz to the upper frequency limit.

Feeder Cable Coaxial cables that run along streets within the served area and connect between the individual taps which serve the customer drops.

Fiber Distributed Data Interface (FDDI) A fiber-based LAN standard.

Fiber Node A point of interface between a fiber trunk and the coaxial distribution.

Forward Channel The direction of RF signal flow away from the head-end toward the end user; equivalent to Downstream.

Frame See MAC frame, S-CDMA frame, and MPEG frame.

Group Delay The difference in transmission time between the highest and lowest of several frequencies through a device, circuit or system.

Guard Time Minimum time allocated between bursts in the upstream referenced from the symbol center of the last symbol of a burst to the symbol center of the first symbol of the following burst. The guard time should be at least the duration of five symbols plus the maximum system timing error.

Harmonic Related Carrier (HRC) A method of spacing television channels on a cable television system in exact 6-MHz increments, with all carrier frequencies harmonically related to a common reference.

Head-end The central location on the cable network that is responsible for injecting broadcast video and other signals in the downstream direction. See also Master Head-End, Distribution Hub.

Header Protocol control information located at the beginning of a protocol data unit.

High Frequency (HF) Used in this document to refer to the entire subsplit (5-30 MHz) and extended subsplit (5-42 MHz) band used in reverse channel communications over the cable television network.

High Return A frequency division scheme that allows bi-directional traffic on a single coaxial cable. Reverse channel signals propagate to the head-end above the downstream passband.

Hum Modulation Undesired modulation of the television visual carrier by the fundamental or low-order harmonics of the power supply frequency, or other low-frequency disturbances.

Hybrid Fiber/Coax (HFC) System A broadband bidirectional shared-media transmission system using fiber trunks between the head-end and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.

Incremental Related Carriers (IRC) A method of spacing NTSC television channels on a cable television system in which all channels except 5 and 6 correspond to the standard channel plan, used to reduce composite triple beat distortions.

Institute of Electrical and Electronic Engineers (IEEE) A voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute.

International Electrotechnical Commission (IEC) An international standards body.

International Organization for Standardization (ISO) An international standards body, commonly known as the International Standards Organization.

Internet Control Message Protocol (ICMP) An Internet network-layer protocol.

Internet Engineering Task Force (IETF) A body responsible, among other things, for developing standards used in the Internet.

Internet Group Management Protocol (IGMP) A network-layer protocol for managing multicast groups on the Internet

Impulse Noise Noise characterized by non-overlapping transient disturbances.

Information Element The fields that make up a MAP and define individual grants, deferred grants, etc.

Internet Protocol (IP) An Internet network-layer protocol.

Interval Usage Code A field in MAPs and UCDs to link burst profiles to grants.

Latency The time, expressed in quantity of symbols, taken for a signal element to pass through a device.

Layer A subdivision of the Open System Interconnection (OSI) architecture, constituted by subsystems of the same rank

Link Layer See Data Link Layer.

Local Area Network (LAN) A non-public data network in which serial transmission is used for direct data communication among data stations located on the user's premises.

Logical Link Control (LLC) procedure In a local area network (LAN) or a Metropolitan Area Network (MAN), that part of the protocol that governs the assembling of data link layer frames and their exchange between data stations, independent of how the transmission medium is shared.

Logical (Upstream) Channel A MAC entity identified by a unique channel ID and for which bandwidth is allocated by an associated MAP message. A physical upstream channel may support multiple logical upstream channels. The associated UCD and MAP messages completely describe the logical channel.

MAC Frame MAC header plus optional PDU.

MAC Service Access Point An attachment to a MAC-sublayer domain.

MAP See Bandwidth Allocation Map.

Master Head-End A head-end which collects television program material from various sources by satellite, microwave, fiber and other means, and distributes this material to Distribution Hubs in the same metropolitan or regional area. A Master Head-End MAY also perform the functions of a Distribution Hub for customers in its own immediate area.

Mean Time to Repair (MTTR) In cable television systems, the MTTR is the average elapsed time from the moment a loss of RF channel operation is detected up to the moment the RF channel operation is fully restored.

Media Access Control (MAC) address The "built-in" hardware address of a device connected to a shared medium.

Media Access Control (MAC) procedure In a subnetwork, that part of the protocol that governs access to the transmission medium independent of the physical characteristics of the medium, but taking into account the topological aspects of the subnetworks, in order to enable the exchange of data between nodes. MAC procedures include framing, error protection, and acquiring the right to use the underlying transmission medium.

Media Access Control (MAC) sublayer The part of the data link layer that supports topology-dependent functions and uses the services of the Physical Layer to provide services to the logical link control (LLC) sublayer.

Micro-reflections Echoes in the forward transmission path due to departures from ideal amplitude and phase characteristics.

Mid Split A frequency division scheme that allows bi-directional traffic on a single coaxial cable. Reverse channel signals propagate to the head-end from 5 to 108 MHz. Forward path signals go from the head-end from 162 MHz to the upper frequency limit. The diplex crossover band is located from 108 to 162 MHz.

Mini-Slot A "mini-slot" is an integer multiple of 6.25-microsecond increments.

Modulation Rate The signaling rate of the upstream modulator (1280 to 5120 kHz). In S-CDMA, the chip rate. In TDMA, the channel symbol rate.

Moving Picture Experts Group (MPEG) A voluntary body which develops standards for digital compressed moving pictures and associated audio.

Multipoint Access User access in which more than one terminal equipment is supported by a single network termination.

Multipoint Connection A connection among more than two data network terminations.

National Cable Television Association (NCTA) A voluntary association of cable television operators which, among other things, provides guidance on measurements and objectives for cable television systems in the USA.

National Television Systems Committee (NTSC) Committee which defined the analog color television broadcast standard used today in North America.

Network Layer Layer 3 in the Open Systems Interconnection (OSI) architecture; the layer that provides services to establish a path between open systems.

Network Management The functions related to the management of data link layer and physical layer resources and their stations across the data network supported by the hybrid fiber/coax system.

Number Of Allocated Codes The total number of codes which a single CM uses in a single S-CDMA frame. This number is determined by the size of the grants in minislots and the mapping of these minislots to S-CDMA frames (note that a CM may receive multiple grants which are mapped to a single S-CDMA frame). The number of allocated codes can be in the range of the number of Codes per Mini-slot to the number of active codes, and may vary from frame to frame, but is constant within an S-CDMA frame.

Open Systems Interconnection (OSI) A framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a service to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.

Organizationally Unique Identifier (OUI) A 3-octet IEEE assigned identifier that can be used to generate Universal LAN MAC addresses and Protocol Identifiers per ANSI/IEEE Std 802 for use in Local and Metropolitan Area Network applications.

Packet Identifier (PID) A unique integer value used to identify elementary streams of a program in a single- or multi-program MPEG-2 stream.

Partial Grant A grant that is smaller than the corresponding bandwidth request from the CM.

Payload Header Suppression The suppression of the header in a payload packet. (*e.g.*, the suppression of the Ethernet header in forwarded packets)

Payload Unit Start Indicator (PUSI) A flag in an MPEG header. A value of 1 indicates the presence of a pointer field as the first byte of the payload.

Physical (PHY) Layer Layer 1 in the Open System Interconnection (OSI) architecture; the layer that provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical and handshaking procedures.

Physical Media Dependent (PMD) Sublayer A sublayer of the Physical Layer which is concerned with transmitting bits or groups of bits over particular types of transmission link between open systems and which entails electrical, mechanical and handshaking procedures.

Primary Service Flow All CMs have a Primary Upstream Service Flow and a Primary Downstream Service Flow. They ensure that the CM is always manageable and they provide a default path for forwarded packets that are not classified to any other Service Flow

Program-Specific Information (PSI) In MPEG-2, normative data necessary for the demultiplexing of Transport Streams and the successful regeneration of programs.

Program Stream In MPEG-2, a multiplex of variable-length digital video and audio packets from one or more program sources having a common time-base.

Protocol A set of rules and formats that determines the communication behavior of layer entities in the performance of the layer functions.

Provisioned Service Flow A Service Flow that has been provisioned as part of the Registration process, but has not yet been activated or admitted. It may still require an authorization exchange with a policy module or external policy server prior to admission.

QoS Parameter Set The set of Service Flow Encodings that describe the Quality of Service attributes of a Service Flow or a Service Class.

Quadrature Amplitude Modulation (QAM) A method of modulating digital signals onto a radio-frequency carrier signal involving both amplitude and phase coding.

Quadrature Phase-Shift Keying (QPSK) A method of modulating digital signals onto a radio-frequency carrier signal using four phase states to code two digital bits.

Radio Frequency (RF) In cable television systems, this refers to electromagnetic signals in the range 5 to 1000 MHz.

Reference Code Matrix A 128-by-128 element matrix formed by stacking successive spreading codes on top of each other, i.e., the bottom row of the reference code matrix is Code 0 (all ones) and the top row is Code 127. The code elements are placed in the matrix from right to left, i.e., the right-most column of the code matrix is the first element of each code, and the left-most column is the last element of each code.

Request For Comments (RFC) A technical policy document of the IETF; these documents can be accessed on the World Wide Web at http://www.rfc-editor.org/.

Return Loss The parameter describing the attenuation of a guided wave signal (*e.g.*, via a coaxial cable) returned to a source by a device or medium resulting from reflections of the signal generated by the source.

Reverse Channel The direction of signal flow towards the head-end, away from the subscriber; equivalent to Upstream.

Routing Information Protocol (RIP) A protocol of the IETF for exchanging routing information about IP networks and subnets.

S-CDMA Frame A two dimensional representation of mini-slots, where the dimensions are codes and time. An S-CDMA frame is composed of p active codes in the code dimension and K spreading intervals in the time dimension. Within the S-CDMA frame, the number of mini-slots is determined by the number of codes per mini-slot (c) and p, the number of active codes in the S-CDMA frame. Each S-CDMA frame thus contains s mini-slots, where s=p/c, and each mini-slot contains c*K information (QAM) symbols.

S-CDMA Subframe A subframe is a vertically-smaller subset of an S-CDMA frame over which interleaving is performed, where the vertical dimension is R' codes, where R' ≤ p (the number of active codes). A subframe is generally used to constrain the interleaving region to be of a similar size to the Reed-Solomon codeword in order to provide protection from impulse noise.

Security Association Identifier A Baseline Privacy security identifier between a CMTS and a CM.

Service Access Point (SAP) The point at which services are provided by one layer, or sublayer to the layer immediately above it.

Service Class A set of queuing and scheduling attributes that is named and that is configured at the CMTS. A Service Class is identified by a Service Class Name. A Service Class has an associated QoS Parameter Set.

Service Class Name An ASCII string by which a Service Class may be referenced in modem configuration files and protocol exchanges.

Service Data Unit (SDU) Information that is delivered as a unit between peer service access points

Service Flow A MAC-layer transport service which:

- Provides unidirectional transport of packets from the upper layer service entity to the RF;
- Shapes, polices, and prioritizes traffic according to QoS traffic parameters defined for the Flow.

Service Flow Identifier (SFID) An identifier assigned to a service flow by the CMTS. [32 bits]

Service Flow Reference A message parameter in Configuration Files and Dynamic Service MAC messages used to associate Classifiers and other objects in the message with the Service Flow Encodings of a requested Service Flow.

Service Identifier (SID) A Service Flow Identifier assigned by the CMTS (in addition to a Service Flow Identifier) to an Active or Admitted Upstream Service Flow. [14 bits]

Simple Network Management Protocol (SNMP) A network management protocol of the IETF.

Spectrum Management System (SMS) A system, defined in [SMS], for managing the RF cable spectrum.

Spread Symbol Or Spreading Interval At the output of the spreader, a group of 128 chips which comprise a single S-CDMA spreading code, and are the result of spreading a single information (QAM) symbol. One spread symbol = one spreading interval = 128 chips = one information (QAM) symbol.

Spreader-Off S-CDMA Burst A transmission from a single CM in a spreader-off frame on an S-CDMA channel defined by the time in which the CM's transmitter turns on to the time it turns off. There will generally be several spreader off bursts in a spreader-off frame.

Spreader-Off S-CDMA Frame TDMA mini-slots on an S-CDMA channel in which the spreader is turned off. These are differentiated from TDMA bursts on a TDMA channel in that, for example, the number of mini-slots per spreader-off SCDMA burst frame is constrained to be the same as the number of mini-slots in a spreader-on SCDMA frame (s). This number of mini-slots will be less than the number of TDMA mini-slots in a TDMA channel over the same time interval if the number of active codes is significantly less than 128.

Spreading Interval Time to transmit a single complete S-CDMA spreading code, equal to the time to transmit 128 chips. Also, time to transmit a single information (QAM) symbol on an S-CDMA channel. See also Spread Symbol.

Sub-Channel A logical channel sharing same upstream spectrum (RF center frequency and RF channel) with other logical channels.

Sublayer A subdivision of a layer in the Open System Interconnection (OSI) reference model.

Subnetwork Subnetworks are physically formed by connecting adjacent nodes with transmission links.

Subnetwork Access Protocol (SNAP) An extension of the LLC header to accommodate the use of 802-type networks as IP networks.

Subscriber See End User.

Subsplit A frequency-division scheme that allows bi-directional traffic on a single cable. Reverse path signals come to the head-end from 5 to 30 (up to 42 on Extended Subsplit systems) MHz. Forward path signals go from the head-end from 50 or 54 MHz to the upper frequency limit of the cable network.

Subsystem An element in a hierarchical division of an Open System that interacts directly with elements in the next higher division or the next lower division of that open system.

System Clock Period The period of the 10.24 MHz system clock, nominally 97.65625 ns.

Systems Management Functions in the application layer related to the management of various open systems Interconnection (OSI) resources and their status across all layers of the OSI architecture.

Tick 6.25-microsecond time intervals that are the reference for upstream mini-slot definition and upstream transmission times.

Tilt Maximum difference in transmission gain of a cable television system over a given bandwidth (typically the entire forward operating frequency range).

Transit Delay The time difference between the instant at which the first bit of a PDU crosses one designated boundary, and the instant at which the last bit of the same PDU crosses a second designated boundary.

Transmission Control Protocol (TCP) A transport-layer Internet protocol which ensures successful end-to-end delivery of data packets without error.

Transmission Convergence Sublayer A sublayer of the Physical Layer that provides an interface between the Data Link Layer and the PMD Sublayer.

Transmission Link The physical unit of a subnetwork that provides the transmission connection between adjacent nodes.

Transmission Medium The material on which information signals may be carried; *e.g.*, optical fiber, coaxial cable, and twisted-wire pairs.

Transmission System The interface and transmission medium through which peer physical layer entities transfer bits.

Transmit On/Off Ratio In multiple-access systems, the ratio between the signal powers sent to line when transmitting and when not transmitting.

Transport Stream In MPEG-2, a packet-based method of multiplexing one or more digital video and audio streams having one or more independent time bases into a single stream.

Trivial File-Transfer Protocol (TFTP) An Internet protocol for transferring files without the requirement for user names and passwords that is typically used for automatic downloads of data and software.

Trunk Cable Cables that carry the signal from the head-end to groups of subscribers. The cables can be either coaxial or fiber depending on the design of the system.

Type/Length/Value (TLV) An encoding of three fields, in which the first field indicates the type of element, the second the length of the element, and the third field the value.

Upstream The direction from the subscriber location toward the head-end.

Upstream Channel Descriptor (UCD) The MAC Management Message used to communicate the characteristics of the upstream physical layer to the cable modems.

4 Abbreviations

ANSI American National Standards Institute

ARP Address Resolution Protocol

ASCII American Standard Code for Information Interchange

ATM Asynchronous Transfer Mode

BPDU Bridge Protocol Data Unit

BPI Baseline Privacy Interface

CA Certificate Authority

CATV Community Access Television, Cable Television

CCCM CPE Controlled Cable Modem

CM Cable Modem

CMCI Cable Modem to CPE Interface

CMTS Cable Modem Termination System

CMTS-NSI Cable Modem Termination System - Network Side Interface

CPE Customer Premises Equipment

CSA Code Signing Agent

CVC Code Verification Certificate

DCC Dynamic Channel Change

DES Digital Encryption Standard

DH Diffie-Helman

DHCP Dynamic Host Configuration Protocol

DOCSIS 1.x Abbreviation for “DOCSIS 1.0 or 1.1”

DOCSIS Data-Over-Cable Service Interface Specifications

ECN Engineering Change Notice

ECO Engineering Change Order

ECR Engineering Change Request

FDDI Fiber Distributed Data Interface

FTP File Transfer Protocol

HFC Hybrid Fiber/Coax (HFC) System

ICMP Internet Control Message Protocol

IE Information Element

IEEE Institute of Electrical and Electronic Engineers

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

IP Internet Protocol

IPCDN Internet Protocol over Cable Data Network (IETF working group)

IPDR Internet Protocol Detail Record

IUC Interval Usage Code

LAN Local Area Network

LLC Logical Link Control procedure

MAC Media Access Control procedure

MIB Management Information Base

MPEG Moving Picture Experts Group

MSAP MAC Service Access Point

MSO Mulitple System Operator

MTA Multimedia Terminal Adapter

NMS Network Management System

OID Object Identifier

OSI Open Systems Interconnection

OSSI Operations Support System Interface

OUI Organization Unique Identifier

PCI Peripheral Component Interconnect

PDU Payload Data Unit

PHS Payload Header Suppression

PHY Physical (PHY) Layer

PID Packet Identifier

PMD Physical Media Dependent (PMD) Sublayer

PSI Program-Specific Information

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase-Shift Keying

RFC Request for Comments

RFI Radio Frequency Interface

RO Read Only

RW Read/Write

SAID Security Association Identifier

SCN Service Class Name

SF Service Flow

SFID Service Flow Identifier

SID Service Identifier

SLA Service Level Agreement

SMI Structure of Management Informationhh

SMS Spectrum Management System

SNAP Subnetwork Access Protocol

SNMP Simple Network Management Protocol

SNMPv1 Version 1 of the Simple Network Management Protocol

SNMPv2c Version 2C of the Simple Network Management Protocol

SNMPv3 Version 3 of the Simple Network Management Protocol

STP Spanning Tree Protocol

TCP Transmission Control Protocol

TFTP Trivial File-Transfer Protocol

TLV Type/Length/Value

UCC Upstream Channel Change

UDP User Datagram Protocol

USB Universal Synchronous Bus

USM User-based Security Model (USM) for version 3 of the Simple Network Management Protocol

VACM View-based Access Control Model for the simple Network Management Protocol (SNMP)

VoIP Voice over Internet Protocol

XML Extensible Markup Language

5 SNMP Protocol[1]

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 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 (Informational)
[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 following IETF SNMP RFC MUST be implemented:

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

5.1 SNMP Mode for DOCSIS 2.0-compliant CMTSes

DOCSIS 2.0-compliant CMTSes MUST support SNMPv1, SNMPv2c, and SNMPv3 and SNMP coexistence as described by [RFC 3411] through [RFC 3415] and [RFC 2576], and MAY support SNMPv1 and SNMPv2c vendor proprietary solutions, including SNMP v1/v2c NmAccess mode, with the following requirements:

1. Revised the matrices and added one below per ECN OSS2-N-03067 by GO on 11/17/03.

a) DOCSIS 2.0 compliant CMTS MUST operate in SNMP coexistence mode (not using docsDevNmAccessTable); additionally, SNMP coexistence mode MAY be disabled, by vendor proprietary configuration control, to allow the CMTS to support SNMPv1, SNMPv2c vendor proprietary solutions, including SNMP v1/v2c NmAccess mode (using docsDevNmAccessTable).

b) CMTSes in SNMPv1/v2c NmAccess mode (using using DOCS-CABLE-DEVICE-MIB docsDevNmAccessTable) MUST operate subject to the following requirements and limitations:

- Only SNMPv1/v2c packets are processed
- SNMPv3 packets are dropped
- The docsDevNmAccessTable controls SNMP access and SNMP trap destinations as described in [RFC 2669]
- None of the SNMPv3 MIBs as defined in [RFC 3411] through [RFC 3415] and [RFC 2576] are accessible.[1]

c) CMTS SNMPv1, SNMPv2c vendor-proprietary solutions MUST operate subject to the following requirements and limitations:

- Only SNMPv1/v2c packets are processed
- SNMPv3 packets are dropped
- Vendor-proprietary solutions MUST control SNMP access and SNMP trap destinations
- None of the SNMPv3 MIBs as defined in [RFC 3411] through [RFC 3415] and [RFC 2576] are accessible.[2]

d) CMTS SNMP Coexistence Mode MUST operate subject to the following requirements and limitations:

- SNMP v1/v2c/v3 Packets are processed as described by in [RFC 3411] through [RFC 3415] 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

5.1.1 Key Change Mechanism

DOCSIS 2.0-compliant CMTSes SHOULD use the key-change mechanism specified in [RFC 2786]. CMTSes MUST always support the key-change mechanism described in [RFC 3414] to comply with the industry wide SNMPv3 standard.

1. Revised this bullet statement per ECN OSS2-N-03014 by GO on 02/26/03.

2. Revised this bullet statement per ECN OSS2-N-03014 by GO on 02/26/03.

5.2 SNMP Mode for DOCSIS 2.0-compliant CMs

DOCSIS 2.0-compliant CMs (in 2.0, 1.1, and 1.0 modes) MUST support SNMPv1, SNMPv2c, and SNMPv3 as well as SNMP-coexistence ([RFC 2576]) subject to 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 such implementation, the access to similar set of MIB objects SHOULD be provided by a diagnostic utility as described in Section 9.3):

- IP connectivity between the CM and the SNMP management station MUST be implemented as described in Section 9.1
- The CM MUST provide read-only access to the following MIB objects:
  docsIfDownChannelFrequency
  docsIfDownChannelPower
  docsIfCmStatusValue
  docsDevServerBootState
  docsDevEventTable[1]
- 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 the configuration file does not contain SNMP access control items (docsDevNmAccessTable or snmpCommunityTable or TLV 34.1/34.2 orTLV38), the CM is in NmAccess mode

1. Added this bullet per ECN OSS2-N-02193 by GO on 02/11/03.

- 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, the modem operates in one of 2 modes. The operating mode is determined by the contents of the config file as described above.

SNMPv1/v2c NmAccess Mode (using docsDevNmAccess Table)

- Only SNMPv1/v2c packets are processed
- SNMPv3 packets are dropped
- docsDevNmAccessTable controls access and trap destinations as described in [RFC 2669]
- None of the SNMPv3 MIBs as defined in [RFC 3411] through [RFC 3415] and [RFC 2576] are accessible.

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] through [RFC 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.
- Notification destinations are specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB.

d) In case of failure to complete SNMPv3 initialization (i.e., NMS cannot 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.

5.2.1 SNMPv3 Initialization and Key changes

DOCSIS 2.0-compliant CMs 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] Annex A and [RFC 3415] Annex A. This means that default usmUser and vacmAccess entries defined in [RFC 3414] Annex A and [RFC 3415] Annex A MUST NOT be present.[1]

5.2.2 SNMPv3 Initialization

For each of up to 5 different security names, the Manager generates a pair of numbers. First, the Manager generates a random number Rm.

Then, the Manager uses the DH equation to translate Rm to a public number z. The equation is as follows:

z = g ^ Rm MOD p

where g is from the set of Diffie-Helman parameters, and p is the prime from those parameters.

The CM configuration file is created to include the (security name, public number) pair. The 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

The CM MUST support the VACM entries defined in Section 5.2.4.

During the CM boot process, the above values (security name, public number) 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
usmUserCloneFrom: 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. The CM generates a random number xa for each row populated in the usmDHKickstartTable which has a non-zero length usmDHKickstartSecurityName and usmDHKickstartMgrPublic.

[1] Added this paragraph per ECN OSS2-N-03067 by GO on 02/19/04.

2. The 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, and p is the prime from those parameters

   At this point:

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

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

4. The 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#5 v2.0.

```
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 2.0-compliant CMs 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 the 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 2.0-compliant CMs MUST provide pre-installed 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.

   The SNMP manager gets the value of the CM’s usmDHKickstartMypublic number associated with the securityName for which the manager wants to derive authentication and privacy keys. Using 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 securityName that the manager is going to use to communicate with the CM.

5.2.3 DH Key Changes

DOCSIS 2.0-compliant CMs MUST support the key-change mechanism specified in [RFC 2786].

5.2.4 VACM Profile

This section addresses the default VACM profile for DOCSIS CMs 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)
  vacmSecurityName: docsisOperator
  vacmGroupName: docsisOperator
  vacmSecurityToGroupStorageType: permanent
  vacmSecurityToGroupStatus: active

- 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 2.0-compliant CMs MUST also support additional VACM users as they are configured via an SNMP-embedded configuration file.

This page intentionally left blank.

6 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.

The DOCSIS OSSI 2.0 specification has priority over the IETF MIBs and all objects. Though deprecated or optional in the IETF MIB, the object can be required by this specification as mandatory. Regardless of having either a status of deprecated or optional in the IETF MIB, the CM and CMTS MUST implement MIB requirements in accordance with the OSSI 2.0 specification.

If not required by this specification, deprecated objects are optional. If a CM or CMTS implements a deprecated object, the object MUST be implemented correctly according to the MIB definition. If a CM or CMTS does not implement a deprecated object, the agent MUST NOT instantiate the object and when accessed, MUST respond with the appropriate error/exception condition, such as no such object for SNMPv2c.

If not required by this specification, optional objects are optional. If a CM or CMTS implements an optional object, the object MUST be implemented correctly according to the MIB definition. If a CM or CMTS does not implement an optional object, the agent MUST NOT instantiate the object and when accessed, MUST respond with the appropriate error/exception condition, such as no such object for SNMPv2c.

If not required by this specification, obsolete objects are optional. If a CM or CMTS implements an obsolete object, the object MUST be implemented correctly according to the MIB definition. If a CM or CMTS does not implement an obsolete object, the agent MUST NOT instantiate the object and when accessed, MUST respond with the appropriate error/exception condition, such as no such object for SNMPv2c.

Section 6.1 and Section 6.2 include an overview of the MIB modules required for management of the facilities specified in the DOCSIS RFI 2.0 and BPI+ specifications.

6.1 IPCDN drafts and others [1]

Table 6-1 IPCDN Drafts

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
[Annex I]	DOCS-LOADBALANCING-MIB	CMTS only

6.2 IETF RFCs[2]

Table 6-2 IETF RFCs

Reference	MIB	Applicable Device(s)
[RFC 2669]	DOCSIS Cable Device MIB: DOCS-CABLE-DEVICE-MIB	CM and CMTS
[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

1. Revised Table 6-1 per ECN OSS2-N-03021(rescinded OSS2-N-02234), OSS2-N-03023, OSS2-N-03069, OSS2-N-03067, and OSSIv2.0-N-04.0126-6 by GO on 03/21/03, 07/11/03, 11/17/03, and 3/15/04.

2. Revised Table 6-2 per ECN OSS2-N-03067 by GO on 11/17/03.

6.3 Managed objects requirements[1]

The following sections detail any additional implementation requirements for the RFCs listed. Refer to Annex 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.

6.3.1 CMTS MIB requirements

DOCSIS 2.0-compliant CMTSes MUST implement the Subscriber Management MIB.

6.3.2 Requirements for [RFC 2669]

[RFC 2669] MUST be implemented by DOCSIS 2.0-compliant CMs. DOCSIS 2.0-compliant CMTSes MUST implement the mandatory required objects (as specified by Annex A), and SHOULD implement the other non-mandatory required objects.

6.3.3 Requirements for DOCS-IF-MIB

DOCS-IF-MIB MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs. It replaces RFC 2670 for DOCSIS 2.0.

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 2.0-conforming CMs. 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.[2]

For modems built based on the multi-programme television distribution used in North-America, 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 across the input power range from -15 dBmV to +15 dBmV.

For modems built based on the European multi-programme television distribution (Euro-DOCSIS), 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 across the input power range from -13 dBmV to +17 dBmV when a 256QAM downstream signal is used, and across the input power range from -17 dBmV to +13 dBmV when a 64QAM downstream signal is used.

---

1. Changed the Titles of Section 6.3.3, 6.3.10, 6.3.11, 6.3.12, 6.3.14, 6.3.15, 6.3.16, 6.3.17, 6.3.18, 6.3.19, 6.3.20, 6.3.21 per ECN OSS2-N-03067 by GO on 11/17/03.

2. Replaced this paragraph and added the two following paragraphs per ECN OSSIv2.0-N-03.0112-2 by GO on 2/19/03.

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.

The docsIfQosProfMaxTransmitBurst range MUST be the same as the one defined in the RFIv2.0 specification, section C.1.1.4.6 "Maximum Upstream Channel Transmit Burst Configuration Setting" which has range 0 to 65535.[1]

In order to be considered a valid modulation profile for assignment to an upstream channel, all entries (IUCs) in the modulation profile MUST have the same value of docsIfCmtsModChannelType. When assigning a modulation profile to an upstream channel, the value of docsIfUpChannelType and the value of docsIfCmtsModChannelType MUST match.

If a modulation profile is in use by one or more upstream channels, the value of docsIfCmtsModChannelType MUST NOT be changed. If a modulation profile is in use by one or more upstream channels, it docsIfCmtsModControl MUST NOT be set to 'destroy' or 'notInService'. Before destroying a modulation profile, or changing the value of docsIfCmtsModChannelType for a profile, the user will need to ensure that it is not currently in use by any upstream channel.

The maximum number of modulation profiles that a CMTS can support in docsIfCmtsModulationTable is vendor -specific.[2]

The CMTS MAY provide pre-defined modulation profiles (entries in the DOCS-IF-MIB docsIfCmtsModulationTable) for the purpose of being used by operators directly or as templates to define other modulation profiles. The pre-defined modulation profiles provided by the CMTS MAY be read-only to prevent users from accidental modifications. It should be noted that adding or creating entries with new docsIfCmtsModIntervalUsageCode values and the same docsIfCmtsModIndex value as a pre-defined modulation profile MAY result in an error.

The modulation profiles are PHY layer specific. Modulation profiles with the same value of docsIfCmtsModIndex may not be optimal for all upstream channels with different physical layer hardware. As a result, re-using modulation profiles for upstream channels with different physical layer hardware could decrease upstream performance. Therefore, SNMP set operations MAY result in an error when modulation profiles with the same value of docsIfCmtsModIndex are assigned to upstream channels with different physical layer hardware.[3]

6.3.4 Requirements for [RFC 2863]

[RFC 2863] MUST be implemented by DOCSIS 2.0-compliant CMTSes 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:

---

1. Added this paragraph per ECN OSS2-N-02221 by GO on 02/11/03.

2. Added the last three paragraphs per ECN OSS2-N-03092 by GO on 11/17/03.

3. Added the last two paragraphs per ECN OSSIv2.0-N-04.0121-3 by GO on 3/3/04.

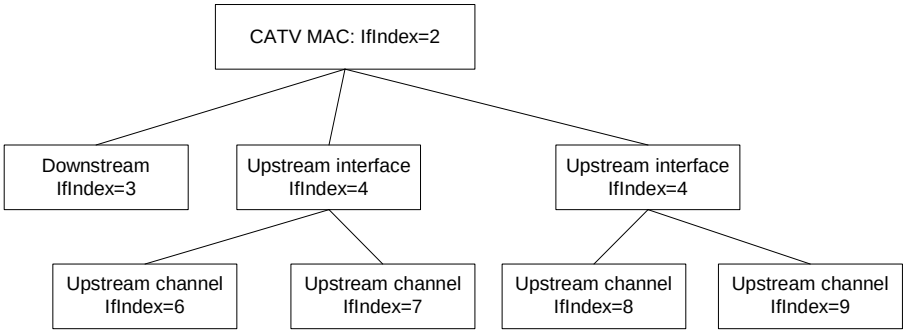
CATV MAC interface: docsCableMacLayer (127)
CATV downstream channel: docsCableDownstream (128)
CATV upstream interface: docsCableUpStream (129)
CATV upstream logical channel: docsCableUpstreamChannel (205)

6.3.4.1 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:

- A CM supports only one upstream interface. At the CM, an instance of IfEntry MUST exist for each CATV-MAC interface, downstream channel, upstream interface, and each LAN interface enabled. The enabling of LAN interfaces MAY be fixed a priori during the manufacturing process or MAY be determined dynamically during operation by the CM according to whether or not an interface has a CPE device attached to it.
- If the CM has multiple CPE interfaces but only one CPE interface can be enabled at any given time, the ifTable MUST contain only the entry corresponding to the enabled or the default CPE interface. If a MAC interface consists of more than one upstream and downstream channel, a separate instance of ifEntry MUST exist for each channel.
- A 2.0 CMTS supports more than one one upstream logical channel per upstream interface. At the CMTS, an instance of IfEntry MUST exist for each CATV-MAC interface, downstream channel, upstream interface, upstream logical channel, and any other interface enabled.
- For CM/CMTS, the ifStack group ([RFC 2863]) must be implemented to identify relationships among sub-interfaces. Note that the CATV-MAC interface MUST exist, even though it is broken out into sub-interfaces.

The following example illustrates a MAC interface with one downstream, two upstream interfaces each with two upstream logical channels for a CMTS.

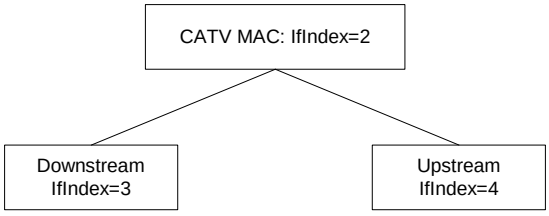


Implementation of ifStackTable for this example:

ifStackHigherLayer	ifStackLowerLayer
0	2
2	3
2	4
2	5
3	0
4	9
4	7
5	9
5	9
6	0
7	0
8	0
9	0

Figure 6-1 ifIndex example for CMTS

The following example illustrates a MAC interface with one downstream and one upstream interface for a CM.



Implementation of ifStackTable for this example:

ifStackHigherLayer	ifStackLowerLayer
0	2
2	3
2	4
3	0
4	0

Figure 6-2 ifIndex example for CM

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, interfaces MUST be numbered as follows:

Table 6-3 CM interface numbering

Interface	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 the CM has more than one CPE interface, the vendor MUST define which of the n CPE interfaces is the primary CPE interface. The definition of the primary CPE interface MAY be fixed a priori 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 of the number of 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 priori 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 CMs 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.

6.3.4.2 docsIfCmStatusValue and ifOperStatus Relationship

For the CM's RF downstream, RF upstream (upstream interface and logical channel) and RF MAC interfaces, the following are the expected relationship of ifOperStatus and docsIfCmStatusValue when ifAdminStatus = up (taken from DOCS-IF-MIB).[1]

Table 6-4 docsIfCmStatusValue and ifOperStatus relationship

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)

1. Revised Table 6-4 per ECN OSS2-N-03065 by GO on 07/11/03.

6.3.4.2.1 ifAdminStatus 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).

6.3.5 Interface MIB and Trap Enable

The Interface MIB and Trap Enable specified in [RFC 2863] MUST be implemented by DOCSIS 2.0-compliant CMTSes 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.

6.3.6 Requirements for [RFC 2665]

[RFC 2665] MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs if Ethernet or Fast Ethernet interfaces are present.

6.3.7 Requirements for [RFC 1493]

[RFC 1493] MUST be implemented by DOCSIS 2.0-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 CMTSes that implement transparent bridging, the Bridge MIB MUST be used to represent information about the MAC Forwarder states.

6.3.8 Requirements for [RFC 2011]

[RFC 2011] MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs.

6.3.8.1 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.

6.3.8.2 The ICMP Group

The ICMP group MUST be implemented.

6.3.9 Requirements for [RFC 2013]

[RFC 2013] MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs.

The UDP group in [RFC 2013] MUST be implemented.

6.3.10 Requirements for [RFC 3418]

[RFC-3418] MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs.

6.3.10.1 The System Group

The System Group from [RFC 3418] MUST be implemented. See Section 7.2.1 for sysObjectID requirements.

6.3.10.2 The SNMP Group

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

6.3.11 Requirements for DOCS-QOS-MIB

Annex J DOCS-QOS-MIB requirements MUST be implemented by DOCSIS 2.0 CMTSes and CMs.[1]

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

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

6.3.13 Requirements for [RFC 2933]

[RFC 2933] MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs.

Refer to Annex E for DOCSIS 2.0 IGMP cable device implementation details.

6.3.14 Requirements for DOCS-BPI2-MIB

"draft-ietf-ipcdn-bpiplus-mib-05.txt" MUST be implemented by DOCSIS 2.0-compliant CMTSes and CMs as specified in Annex A.[2]

---

1. Added this sentence to the section per ECN OSSIv2.0-N-04.0127-4 by GO on 3/16/04.

6.3.15 Requirements for USB MIB

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

6.3.16 Requirements for DOCS-SUBMGT-MIB

“draft-ietf-ipcdn-subscriber-mib-02-.txt” MUST be implemented by DOCSIS 2.0-compliant CMTSes.

DOCSIS 2.0-compliant CMTSes MUST support a minimum number of thirty (30) filter groups of twenty (20) filters each.

6.3.17 Requirements for [RFC 2786]

[RFC 2786] MUST be implemented by DOCSIS 2.0-compliant CMs. [RFC 2786] MAY be implemented on the CMTS.

6.3.18 Requirements for [RFC 3083]

[RFC 3083] MUST be implemented by DOCSIS 2.0-compliant CMs as specified in Annex A.

Due to the editorial error in [RFC 3083], DOCSIS 2.0-compliant CMs 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.

---

2. Revised sentence (rescinded OSS2-N-02234) per ECN OSS2-N-03021 by GO on 03/21/03.

6.3.19 Requirements for DOCS-IF-EXT-MIB

A DOCSIS 2.0-compliant CM/CMTS MUST NOT support the DOCS-IF-EXT MIB, which is defined in Annex G.

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

DOCSIS 2.0-compliant CMs and CMTSes MUST implement DOCS-CABLE-DEVICE-TRAP-MIB, as specified in Annex H.

6.3.21 Requirements for SNMPv3 MIBs

DOCSIS 2.0-compliant CMs/CMTSes MUST implement the MIBs defined in [RFC 3411] through [RFC 3415] and [RFC 2576][1].

For CMs, the default value for any SNMPv3 object with a storageType textual convention MUST be 'volatile(2)'. This overides the default value specified in [RFC 3411] through [RFC 3415] and [RFC 2576]. The CM MUST only accept the value of 'volatile(2)' on any SNMPv3 storageType object. An attempted set to a value of other(1), nonVolatile(3), permanent(4), or readOnly(5) will result in an 'inconsistantValue' error. Values other than the valid range (1-5) would result in a 'wrongValue' error.

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

6.3.22 Requirements for DOCS-LOADBALANCING-MIB[2]

DOCSIS 2.0-compliant CMTSes MUST implement DOCS-LOADBALANCING-MIB, as specified in Annex I.

6.4 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 Annex C of [DOCSIS 5].

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

TLV-11 translation defines the process used by the CM to convert CM configuration file information (TLV-11 elements) into SNMP PDU (varbinds). The CM MUST translate 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 processes the SNMP PDU and determines the CM configuration pass/fail based on the rules for CM configuration file processing, described below. However, if a CM is not physically

1. Revised this sentence per ECN OSS2-N-03067 by GO on 4/5/04.

2. Added new section per ECN OSSIv2.0-N-04.0126-6 by GO on 3/15/04.

capable of processing a potentially large single CM configuration file-generated SNMP PDU, 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], 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 CM configuration file MUST NOT contain duplicate TLV-11 elements (duplicate means SNMP MIB object has identical OID). If duplicate TLV-11 elements are received by the CM, from the CM configuration file, then the CM MUST fail CM configuration.[1]

6.4.1.1 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.

6.4.2 CM configuration TLV-11 elements not supported by the 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 vendors' 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 treat 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 Section 7.4.2.3.

6.4.3 CM state after CM configuration file processing success

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

---

1. Replaced paragraph per ECN OSSIv2.0-N-03.0115-2 by GO on 2/19/04.

6.4.4 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, 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.

6.5 Treatment and interpretation of MIB counters on the CM

Octet and packet counters implemented as counter32 and counter64 MIB objects are 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 Section 3.1.6 of [RFC 2863]).

6.6 SNMPv3 Notification Receiver config file element

This section specifies processing requirements on the CM when one or SNMPv3 Notification Receiver TLVs are present in the configuration file. The SNMPv3 Notification Receiver TLV is used to configure SNMPv3 tables for notification transmission. The CM MUST process this TLV only if the CM is in SNMPv3 Coexistence Mode.

Based on the TLV, the CM MUST make entries to the following tables in order to cause the desired trap transmission: snmpNotifyTable, snmpTargetAddrTable, snmpTargetParamsTable, snmpNotifyFilterProfileTable, snmpNotifyFilterTable, snmpCommunityTable, usmUserTable, vacmSecurityToGroupTable, vacmAccessTable, and vacmViewTreeFamilyTable. The mapping from the TLV to these tables is described in the following section.

6.6.1 Mapping of TLV fields into created SNMPv3 table rows

The following tables illustrate how the fields from the config file TLV elements are placed into the SNMPv3 tables. The TLV fields are shown below as:

<IP Address> A 32-bit IP address of a notification receiver

<Port> A 16-bit UDP Port number on the notification receiver to receive the notifications

<Trap type> Defines the notification type as explained above

<Timeout> 16-bit timeout, in milliseconds to wait before sending a retry of an Inform Notification

<Retries> 16-bit number of times to retry an Inform after the first Inform transmission

<Filter OID> The OID of the snmpTrapOID value that is the root of the MIB subtree that defines all of the notifications to be sent to the Notification Receiver.

<Security Name> The security name specified on the TLV element, or "@config" if not specified.

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

In configuring entries in these SNMPv3 tables, note the following:

- The Community Name for traps in SNMPv1 and SNMPv2 packets is configured as "public". The Security Name in traps and informs in SNMPv3 packets where no security name has been specified is configured as "@Config", in which case the security level is "noAuthNoPriv".
- Several columnar objects are configured with a value beginning with the string "@config". If these tables are configured through other mechanisms, Network operators should not use values beginning with "@config" to avoid conflicts with the mapping process specified here.

6.6.1.1 snmpNotifyTable

The snmpNotifyTable is defined in [RFC 3413], in the "Notification MIB Module" section.

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

Table 6-5 snmpNotifyTable

Column Name (* = Part of Index)	1st Row Column Value	2nd Row Column Value
* snmpNotifyName	"@config_inform"	"@config_trap"
snmpNotifyTag	"@config_inform"	"@config_ trap "
snmpNotifyType	inform (2)	trap (1)
snmpNotifyStorageType	volatile (2)	volatile (2)
snmpNotifyRowStatus	Active (1)	active (1)

6.6.1.2 snmpTargetAddrTable

The snmpTargetAddrTable is defined in [RFC 3413], in the "Management Target MIB Module" section.

Create 1 row for each TLV element in the config file.

Table 6-6 snmpTargetAddrTable

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	"@config_trap" if <Trap type> is 1, 2, or 4 "@config_inform" if <Trap type> is 3 or 5
snmpTargetAddrParams	"@config_n" (Same as snmpTargetAddrName value)
snmpTargetAddrStorageType	volatile (2)
snmpTargetAddrRowStatus	active (1)

6.6.1.3 snmpTargetAddrExtTable

The snmpTargetAddrExtTable is defined in [RFC 3413], in the "SNMP Community MIB Module" section.

Create 1 row for each TLV element in the config file.

Table 6-7 snmpTargetAddrExtTable

Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@config_n" where n ranges from 0 to m-1, and m is the number of notification receiver TLV elements in the config file
snmpTargetAddrTMask	<zero-length octet string>
snmpTargetAddrMMS	0

6.6.1.4 snmpTargetParamsTable

The snmpTargetParamsTable is defined in [RFC 3413], in the "Management Target MIB Module" section.

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 6-8 snmpTargetParamsTable for <Trap type> 1, 2, or 3

Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" where n ranges from 0 to m-1, and m is the number of notification receiver TLV elements in the config file
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	SNMPv1 (0) if <Trap type> is 1 SNMPv2c (1) if <Trap type> is 2 or 3 SNMPv3 (3) if <Trap type> is 4 or 5

Table 6-8 snmpTargetParamsTable for <Trap type> 1, 2, or 3

Column Name (* = Part of Index)	Column Value
snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	SNMPv1 (1) if <Trap type> is 1 SNMPv2c (2) If <Trap type> is 2 or 3 USM (3) if <Trap type> is 4 or 5 NOTE: The mapping of SNMP protocol types to value here are different from snmpTargetParamsMPModel
snmpTargetParamsSecurityName	"@config"
snmpTargetParamsSecurityLevel	noAuthNoPriv
snmpTargetParamsStorageType	volatile (2)
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 6-9 snmpTargetParamsTable for <Trap type> 4 or 5

Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" where n ranges from 0 to m-1, and m is the number of notification receiver TLV elements in the config file
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	SNMPv1 (0) if <Trap type> is 1 SNMPv2c (1) if <Trap type> is 2 or 3 SNMPv3 (3) if <Trap type> is 4 or 5
snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	SNMPv1 (1) if <Trap type> is 1 SNMPv2c (2) if <Trap type> is 2 or 3 USM (3) if <Trap type> is 4 or 5 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 (2)
snmpTargetParamsRowStatus	active (1)

6.6.1.5 snmpNotifyFilterProfileTable

The snmpNotifyFilterProfileTable is defined in [RFC 3413], in the "Notification MIB Module" section.

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

Table 6-10 snmpNotifyFilterProfileTable

Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" where n ranges from 0 to m-1, and m is the number of notification receiver TLV elements in the config file
snmpNotifyFilterProfileName	"@config_n" where n ranges from 0 to m-1, and m is the number of notification receiver TLV elements in the config file
snmpNotifyFilterProfileStorType	volatile (2)
snmpNotifyFilterProfileRowStatus	active (1)

6.6.1.6 snmpNotifyFilterTable

The snmpNotifyFilterTable is defined in [RFC 3413], in the "Notification MIB Module" section.

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

Table 6-11 snmpNotifyFilterTable

Column Name (* = Part of Index)	Column Value
* snmpNotifyFilterProfileName	"@config_n" where n ranges from 0 to m-1, and 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 (2)
snmpNotifyFilterRowStatus	active (1)

6.6.1.7 snmpCommunityTable

The snmpCommunityTable is defined in [RFC 3413], in the "SNMP Community MIB Module" section.

Create 1 row with fixed values if 1 or more TLVs are present. This causes SNMPv1 and v2c notifications to contain the community string in snmpCommunityName.

Table 6-12 snmpCommunityTable

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 (2)
snmpCommunityStatus	active (1)

6.6.1.8 usmUserTable

The usmUserTable is defined in [RFC 3414], in the "Definitions" section.

Create 1 row with fixed values if 1 or more TLVs are 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 which notifications are to be sent.

One row in the usmUserTable is created. 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 6-13 usmUserTable

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> This row cannot be cloned.
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 (2)
usmUserStatus	active (1)

6.6.1.9 vacmSecurityToGroupTable

The vacmSecurityToGroupTable is defined in [RFC 3415], in the "Definitions" section.

Create 3 rows with fixed values if 1 or more TLVs are 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>. 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.

Table 6-14 vacmSecurityToGroupTable

Column Name (* = Part of Index)	First Row Column Value	Second Row Column Value	Third Row Column Value
* vacmSecurityModel	SNMPV1 (1)	SNMPV2c (2)	USM (3)
* vacmSecurityName	"@config"	"@config"	"@config"
vacmGroupName	"@configV1"	"@configV2"	"@configUSM"
vacmSecurityToGroupStorageType	volatile (2)	volatile (2)	volatile (2)
vacmSecurityToGroupStatus	active (1)	active (1)	active (1)

6.6.1.10 vacmAccessTable

The vacmAccessTable is defined in [RFC 3415], in the "Definitions" section.

Create 3 rows with fixed values, if 1 or more TLVs are 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>. 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.

Table 6-15 vacmAccessTable

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 (2)	volatile (2)	volatile (2)
vacmAccessStatus	active (1)	active (1)	active (1)

6.6.1.11 vacmViewTreeFamilyTable

The vacmViewTreeFamilyTable is defined in [RFC 3415], in the "a" section.

Create 1 row with fixed values if 1 or more TLVs are 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>. 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.

Table 6-16 vacmViewTreeFamilyTable

Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	"@config"
* vacmViewTreeFamilySubtree	1.3
vacmViewTreeFamilyMask	<default from MIB>
vacmViewTreeFamilyType	included (1)
vacmViewTreeFamilyStorageType	volatile (2)
vacmViewTreeFamilyStatus	active (1)

This page intentionally left blank.

7 OSSI for Radio Frequency Interface

7.1 Subscriber Account Management Interface Specification[1]

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 Annex B.

7.1.1 Service Flows, Service Classes, and Subscriber Usage Billing

The DOCSIS 2.0 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.

1. Replaced Section 7.1 per ECN OSS2-N-02236 by GO on 02/12/03.

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 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[1] 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 2.0 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.

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

7.1.1.1 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.[1] 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 re-initializes its management interface. The CMTS sysUpTime value is used to determine if the management interface has been reset between adjacent collection intervals. It is expected that the 64-bit counters will not roll over within the service lifetime of the CMTS.

---

1. 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.

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 Annex 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. Loss-less 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 7.1.4.
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 7.1.5.

7.1.2 IP Detail Record (IPDR) Standard

The IPDR Organization (see http://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 Annex A for the DOCSIS OSSI Service Specification submission to IPDR.org for the DOCSIS Cable Data Systems Subscriber Usage Billing Record. Annex 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.

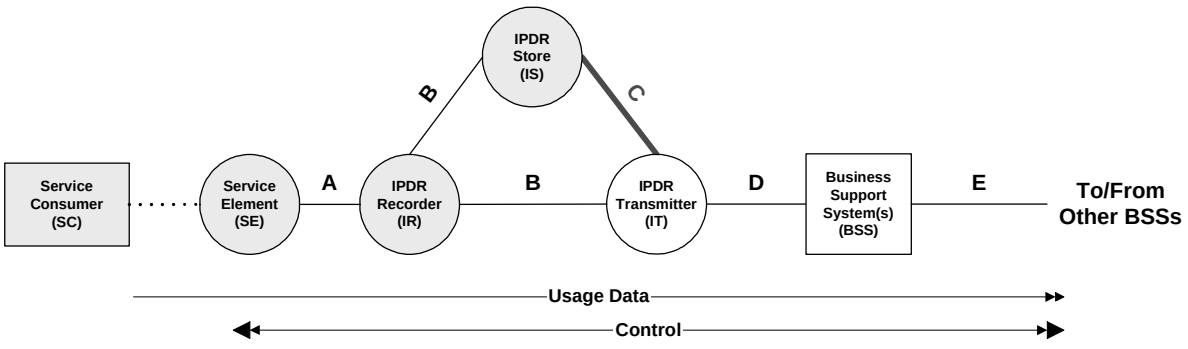


Figure 7-1 Basic Network Model (ref. [NDM-U 3.1] from www.ipdr.org)

7.1.2.1 IPDR Network Model

The IPDR Network Model is given in the [NDM-U 3.1] specification and is portrayed in Table 7-1 above. Note that in Table 7-1 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 7.1.4. In this specification the IT retrieves the compressed and possibly encrypted billing file from the IS on a collection cycle determined by the IT.

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 Annex E). In addition, the billing file in the C-interface is compressed as required by Section 7.1.1. 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 Annex B. The D- and E-interfaces are beyond the scope of this specification.

7.1.2.2 IPDR Record Structure

The [NDM-U 3.1] specification specifies the IPDRDoc record structure. The IPDRDoc XML schema (see IPDRDoc3.1.xsd in Annex B) defines the hierarchy of elements within the IPDR document that MUST be supported by the CMTS as shown in Figure 7-2 below.

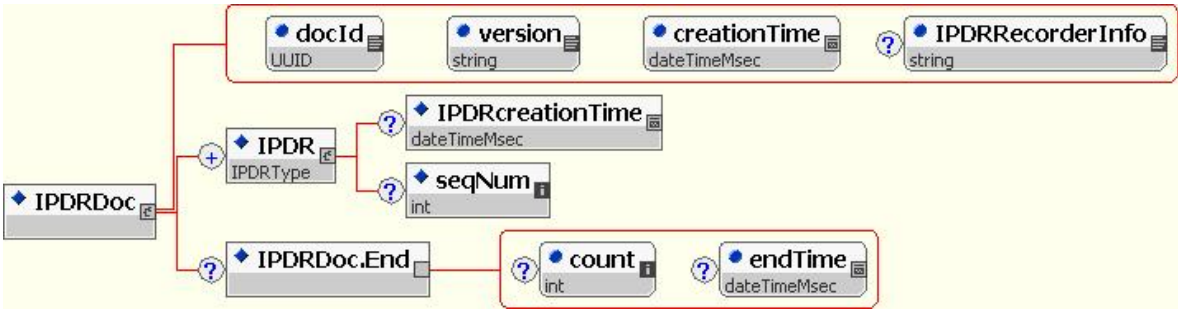


Figure 7-2 IPDRDoc 3.1 Generic Schema

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 Annex E) defines the elements that record the DOCSIS specific information that MUST be supported by the CMTS (as shown in Figure 7-3 below). Note that the DOCSIS-Type in Figure 7-3 is the application specific implementation of the IPDR element shown in Figure 7-3. Thus, the DOCSIS specific elements are sub elements of the IPDR element.

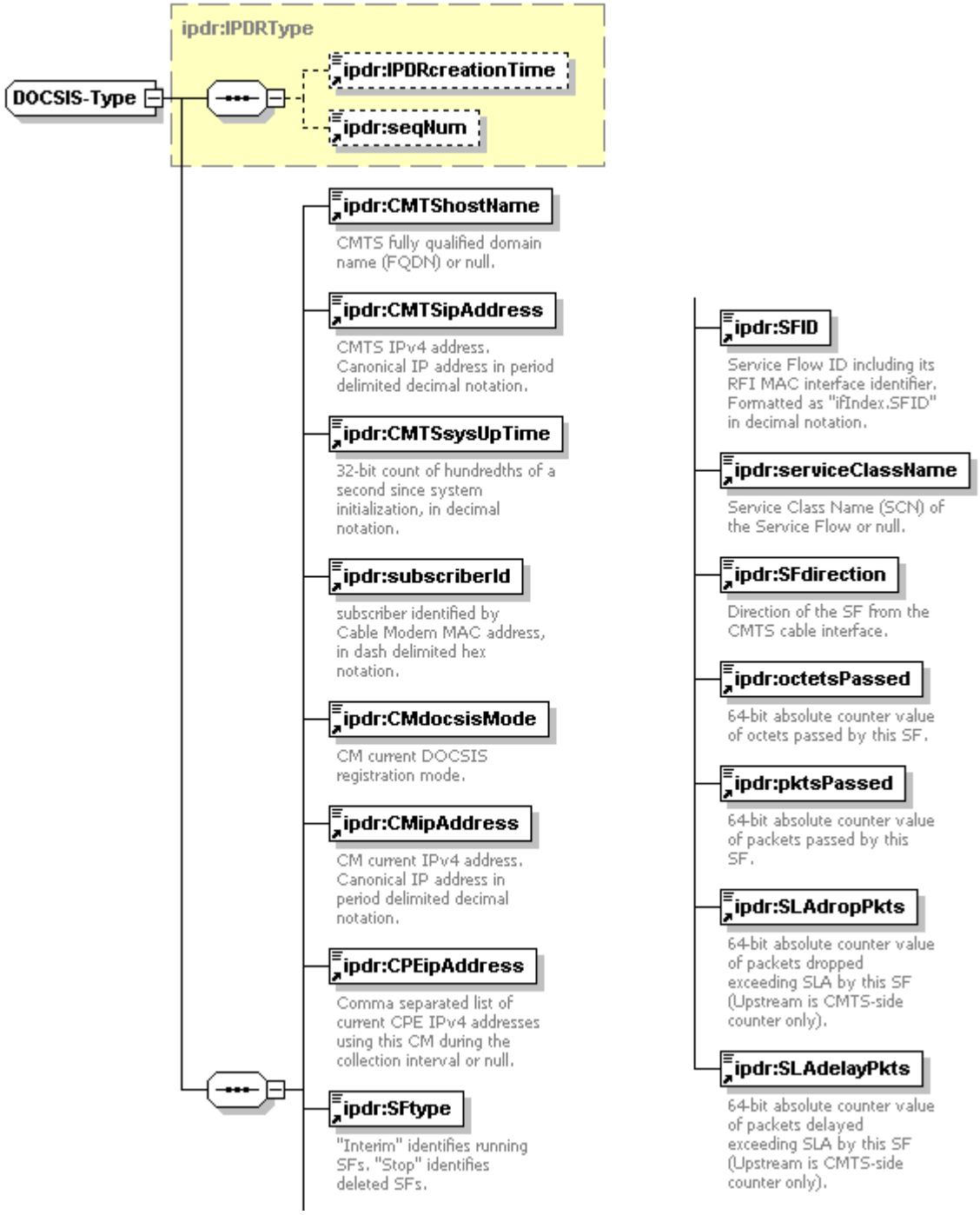


Figure 7-3 DOCSIS IPDR 3.1 Schema

Note: The following elements and attributes are the only ones used by the DOCSIS Cable Data Systems Subscriber Usage Billing Record IPDR instance document (see Annex B). 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 CMTSes 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 1. 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

    a) 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".

7.1.3 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 7-4 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 7-4 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.

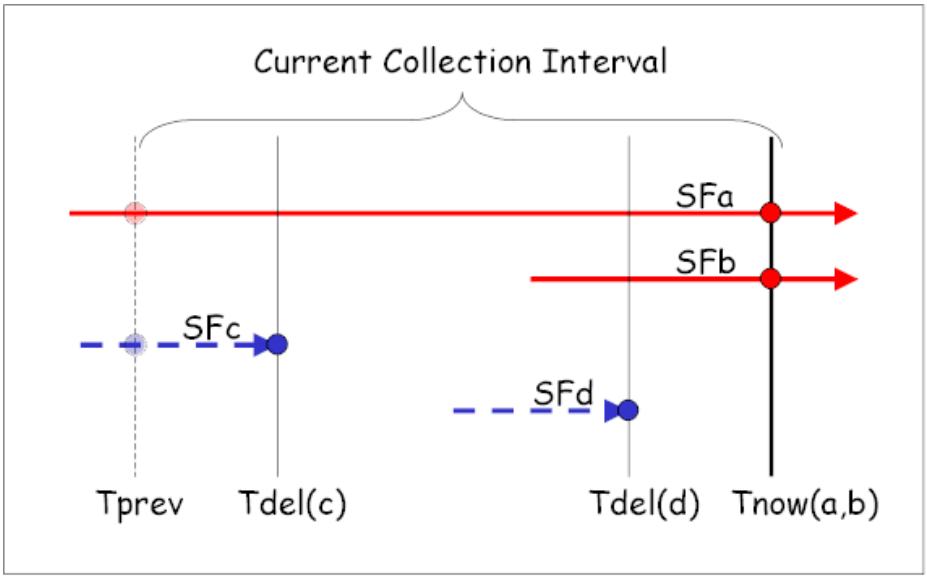


Figure 7-4 Billing Collection Interval Example

7.1.4 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.

7.1.5 Billing File Security Model

The billing file security model has two components: 1) secure 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 network access to the formatter's platform via legacy insecure Telnet and FTP when SSH2/SFTP are active. The billing collector MUST have it's own userid and password for access to the billing file directory via SSH2/SFTP and this 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. However, it is strongly recommended that the operator not allow insecure legacy FTP access to the formatter's billing file.

7.2 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 *non-operational* or *operational*.

Non-operational changes occur when a manager issues a modify command to a CM/CMTS, and the change doesn't affect 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 6 of this document.

While the network is in operation, configuration management is 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.

7.2.1 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>

To report	Format of each field
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 sysObjectID and sysDescr is to define how to report information in a consistent manner so that sysObjectID and 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 the CMTS, the format and content of the information in sysDescr is vendor-dependent.

7.2.2 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.

7.2.3 Secure Software Upgrades

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

DOCSIS 2.0 CMs MUST use the secure software upgrade mechanism to perform software upgrade regardless of the version (1.0, 1.1, or 2.0) of the CMTS to which it is connected.

When a 2.0 CM is connected to a 2.0 CMTS, the CM MUST operate in either DOCSIS 2.0 mode, DOCSIS 1.1 mode, or DOCSIS 1.0 mode.

When a 2.0 CM is connected to a 1.1 CMTS, the CM MUST operate in either DOCSIS 1.1 mode or DOCSIS 1.0 mode.

When a 2.0 CM is connected to a 1.0 CMTS, the CM MUST operate in DOCSIS 1.0 mode.

This means that a DOCSIS 2.0 CM MUST use secure software upgrade mechanism to perform software upgrade regardless of what mode it operates in (1.0 mode, 1.1 mode or 2.0 mode). There are two available secure software download schemes: the manufacturer control scheme and the operator control scheme.

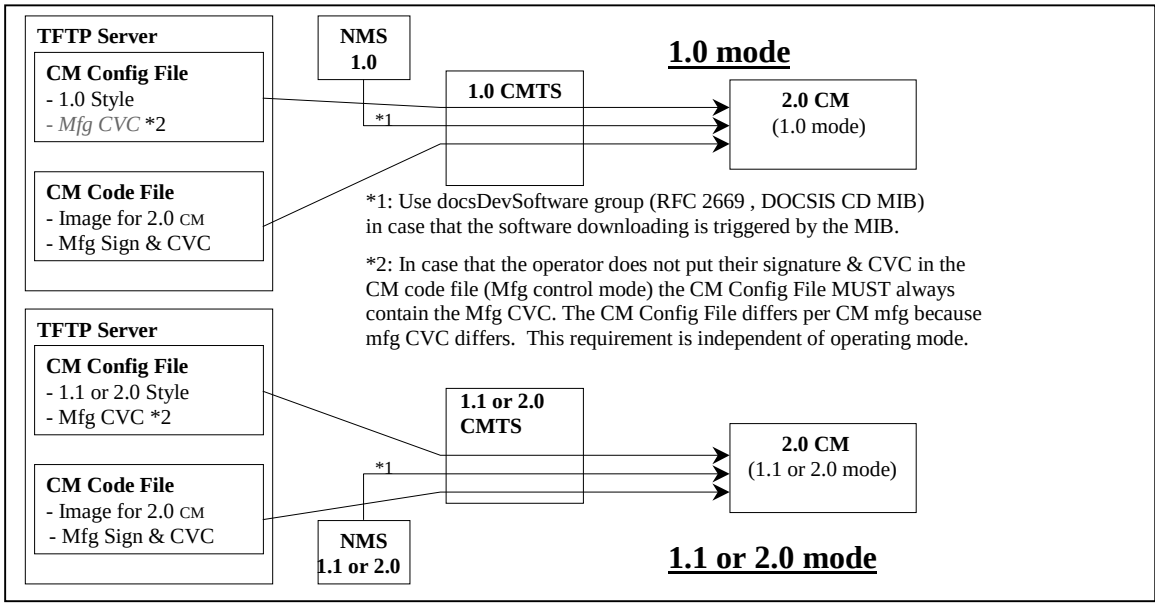


Figure 7-5 Manufacturer control scheme

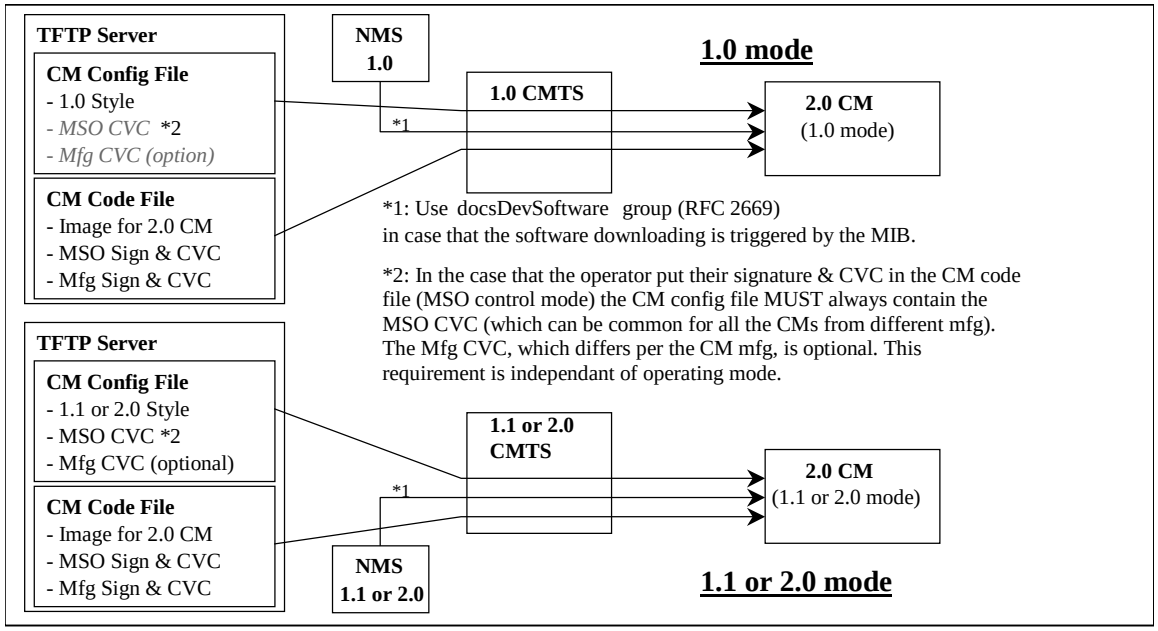


Figure 7-6 Operator control scheme

Prior to secure software upgrade initialization, CVC information needs 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 included 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 a SNMP-initiated upgrade as a method to trigger a secure software upgrade operation. If the operator decides to use a configuration-file-initiated upgrade as a method to trigger secure software download, CVC information needs 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: SNMP-initiated and configuration-file-initiated. Both methods MUST be supported by CMs and MAY be supported by CMTSes.

The following describes the SNMP-initiated mechanism. Prior to a SNMP-initiated 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.[1]

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}
- ocsDevSwFilename 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}
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the CM
- docsDevOperStatus MUST be completeFromMgt{3}

1. Replaced this paragraph and the following paragraphs to the end of the section per ECN OSSIv2.0-N-03.0108-2 by GO on 2/19/03.

- 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 in Progress{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 12.1 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 12.1 of [DOCSIS 5].)
- docsDevSwOperStatus MUST be in Progress{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:

- docDevSwAdminStautus 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 12.1 "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 Annex D, 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

7.3 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.

7.3.1 LLC filters

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.

7.3.1.1 docsDevFilterLLCUnmatchedAction

If the 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 the inbound traffic direction only. Traffic generated from the 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.

7.3.2 Special filters

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

7.3.3 IP spoofing filter

DOCSIS 2.0 CMs MAY implement an IP spoofing filter as 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
- The default value of docsDevCpeIpMax = -1

7.3.3.1 Additional requirement on dot1dTpFdbTable [RFC 1493]

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

7.3.4 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 the CM is running in SNMP Coexistence mode, SNMP access MUST be controlled and specified by the MIB Objects in [RFC 3411] through [RFC 3415], and [RFC 2576].

7.3.4.1 docsDevNmAccessIP and docsDevNmAccessIpMask

A device that implements docsDevNmAccessTable applies the following rules in order to determine whether to permit SNMP access from a given source IP address (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 docsDevNmAccesIpMask)), the CMTS/CM MUST permit the access from SrcIpAddr.
3. If neither #1 nor #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 table contains sample MIB values and the access granted.

Table 7-1 Sample docsDevNmAccessIp values

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

7.3.5 IP filter

The object docsDevFilterIPDefault MUST apply to all CM interfaces. DOCSIS 2.0-compliant CMs MUST support a minimum 16 IP filters.

7.4 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 (such as ability to trace and identify faults, accept and act on error-detection events), as well as the ability to diagnose and correct problems (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, and diagnosis and correction of problems.

7.4.1 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 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/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.[1]

The cable device (CMTS) MUST implement the extended version of MIB object docsIfCmtsCmStatusValue of ([RFI-MIB-IPCDN-DRAFT]) as follows:[1]

```
docsIfCmtsCmStatusValue OBJECT-TYPE
   SYNTAX      INTEGER {
            other(1),
            ranging(2),
            rangingAborted(3),
            rangingComplete(4),
            ipComplete(5),
            registrationComplete(6),
            accessDenied(7),
            operational(8), --deprecated
            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-IO5-040407,
     Section 11.2."
   ::= { docsIfCmtsCmStatusEntry 9 }
```

1. Revised paragraph per ECN OSS2-N-03069 by GO on 07/11/03.

1. Added this paragraph and the subsequent related text per ECN OSS2-N-03069 by GO on 07/11/03.

The cable device (CMTS) MUST 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 MUST implement the extended version of MIB object docsIfCmtsModPreambleType of DOCS-IF-MIB as follows:[1]

```
docsIfCmtsModPreambleType        OBJECT-TYPE
    SYNTAX       INTEGER {
            unknown(0),
        qpsk0(1),
        qpsk1(2)
        }
    MAX-ACCESS  read-create
    STATUS              current
    DESCRIPTION
        "Preamble type for DOCSIS 2.0 bursts. The value 'unknown(0)' represents a row entry
         consisting only of DOCSIS 1.x bursts."
    REFERENCE
        " Data-Over-Cable Service Interface Specifications: Radio
        Frequency Interface Specification SP-RFIv2.0-IO5-040407,
        Section 8.3.3 Upstream Channel Descriptor (UCD),
        Table 8-19 and section 6.2.9 Preamble Prepend."
    DEFVAL { qpsk0 }
    ::= { docsIfCmtsModulationEntry 16 }
```

The cable device CMTS MUST implement the extended version of MIB object docsIfCmtsChannelUtId of DOCS-IF-MIB as follows: [2]

```
docsIfCmtsChannelUtId OBJECT-TYPE
        SYNTAX            Integer32 (0..255)
        MAX-ACCESS  not-accessible
        STATUS              current
        DESCRIPTION
             "The tertiary index into this table. Indicates the CMTS
     identifier for this physical channel."
        ::= { docsIfCmtsChannelUtilizationEntry 2 }
```

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):

1. Added this sentence and subsequent related information per ECN OSS2-N-03087 by GO on 11/17/03.

2. Added this sentence and subsequent related information per ECN OSS2-N-03091 by GO on 11/17/03.

```
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.

7.4.2 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 the cable device's SNMP mode):

- 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 the current specification. 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 the current specification, and SNMP notification functions as specified in [RFC 3413].

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 the current specification; however, SNMP TRAP (trap-versions/targets) MAY be implemented as specified in [RFC 2669] and OSSI 1.1, or a vendor-proprietary MIB. When the CMTS is in SNMP Coexistence 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 the current specification, and SNMP notification functions as specified in [RFC 3413].

7.4.2.1 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 2.0 compliant cable device (CM/CMTS) MUST support the docsDevEvControlTable with additional requirements as described in this specification.[1]

The cable device event log MUST be organized as a cyclic buffer with a minimum of ten entries, and MAY persist across reboots. The event log table MUST be accessible through the DocsDevEventTable [RFC 2669] by the cable device (CM or CMTS).

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.

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 Annex D.

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

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

Section 7.4.2.2.2 describes rules to generate unique EventIds from the error code.

The [RFC 2669] docsDevEvIndex object 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 the [RFC 2669] docsDevEvControl object MUST clear both the local-volatile and local-nonvolatile logs.

1. Revised this paragraph per ECN OSS2-N-03034, by GO on 04/21/03.

7.4.2.2 Format of Events

Annex D lists all DOCSIS events.

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

7.4.2.2.1 SNMP TRAP/INFORM

A cable device (CMTS or CM) MUST send the following generic SNMP traps, as defined in standard MIB [RFC 3418] and [RFC 2863]:

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

A cable device (CMTS or 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, and DOCS-IF-MIB) and defined in Annex H:

- A CM or CMTS in SNMP v1/v2c NmAccess mode MUST support SNMPv1 and SNMPv2c Traps.
- A CM or CMTS in SNMP Coexistence mode MUST support SNMPv1, SNMPv2c, and SNMPv3 Traps.
- Cable devices (CMTS or CM) MUST support INFORM.

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 an 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 Table 7-2. 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 Table 7-3. It MAY send (optionally) events with other priorities.

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

When defining a vendor-specific SNMP trap, the OBJECTS statement of the private trap definition SHOULD contain at least the objects explained below. For CM traps, docsDevEvLevel, docsDevEvId, docsDevText, docsIfDocsisCapability, docsIfDocsisCapability, ifPhysAddress, and docsIfCmCmtsAddress SHOULD be included. For CMTS traps, docsDevEvLevel, docsDevEvId, docsDevEvText, docsIfCmtsCmStatusDocsisMode, docsIfCmtsCmStatusMacAddress, docsIfDocsisOperMode, and ifPhysAddress SHOULD be included. For a description of the usage of these objects, please refer to the DOCS-CABLE-DEVICE-TRAP-MIB. 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 local event logging on a particular priority whenever the SNMP traps are configured on that event priority.

7.4.2.2.2 SYSLOG message format

For DOCSIS events, the CM's Syslog message MUST be sent in the following format; for non-DOCSIS events, it is optional:

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

Where *level* is an ASCII representation of the event priority, enclosed in angle brackets, which is constructed as an OR of the default Facility (128) and event priority (0-7). The resulting level has the range between 128 and 135.

The CMTS's Syslog message MUST be sent in the following format:

For DOCSIS events, the CMTS's Syslog message MUST be sent in the following format; for non-DOCSIS events, it is optional:

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

Where:

- *level* is an ASCII representation of the event priority, enclosed in angle brackets, which is constructed as an OR of the default Facility (128) and event priority (0-7). The resulting level ranges between 128 and 135.
- *TIMESTAMP* and *HOSTNAME* MAY be sent after <level> by the CMTS. If the *TIMESTAMP* and *HOSTNAME* fields are sent, they MUST be in the same format as in Section 4.1.2 of [RFC 3410]. *TIMESTAMP* and *HOSTNAME* format and MUST be sent together. The one space after *TIMESTAMP* is part of the *TIMESTAMP* field. The one space after *HOSTNAME* is part of the *HOSTNAME* field.[1]
- *vendor* is the vendor name for the vendor-specific SYSLOG messages or DOCSIS for the standard DOCSIS messages.
- *eventId* is an ASCII representation 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 the docsDevEvId object in docsDevEventTable, and is also associated with SNMP TRAP in the "SNMP TRAP/Inform" section.

  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.
  - The 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 gap 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 gap in the left side.

  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 contain the textual description as defined in Annex D.
- *vendor-specific-text* MAY be provided by vendors for vendor-specific information.

---

1. Revised this bullet per ECN OSS1v2.0-N-03.0117-2 by GO on 2/19/04.

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.

For example, the syslog event for the event D04.2, "Time of the day received in invalid format", is as follows:

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

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

7.4.2.3 Standard DOCSIS events for CMs

The DOCS-CABLE-DEVICE-MIB [RFC 2669] defines 8 priority levels and a corresponding reporting mechanism for each level.

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 their own set of emergency events. Examples of such events might be 'no memory buffers available', 'memory test failure', and so on. (Such basic cross-vendor type events should be included in the DOCSIS 2.0 "Events for Notification" Appendix F so that vendors do not define many overlapping EventIds in vendor-private scope.)

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, the system MUST send a cold/warm start notification. The 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 the docsIfCmStatusCode SNMP variable of 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 of DOCS-IF-MIB. Examples of such events might be configuration file problems detected by the modem or the inability to get an IP address from the DHCP server.

Error event (priority 4) A failure occurred that could interrupt the normal data flow but will not cause the modem to re-register. Error events could be reported in real time by using the trap or SYSLOG mechanism.

Warning event (priority 5) A failure occurred that could interrupt the normal data flow but will not cause the modem to re-register. 'Warning' level is assigned to events both modem and CMTS have information about. To prevent sending the same event both from the CM and the CMTS, the trap and Syslog reporting mechanism is disabled by default for this level.

Notice event (priority 6) The event is important, but is not a failure and could be reported in real time by using the trap or SYSLOG mechanism. For a CM, an example of a Notice event is 'SW UPGRADE SUCCESS'.[1]

Informational event (priority 7) The event is of marginal importance, and 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 5.2 of this document.[2]

---

1. Revised last sentence per ECN OSS2-N-02193 by GO on 02/11/03.

Debug event (priority 8) Reserved for vendor-specific non-critical events.

The reporting mechanism for each priority could be changed from the default reporting mechanism Table 7-2 by using docsDevEvReporting object of DOCS-CABLE-DEVICE-MIB.[1] [2]

Table 7-2 Default event priorities for the Cable Modem device

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

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

7.4.2.4 Standard DOCSIS events for CMTSes

CMTSes 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 the CMTS device. In all these cases the CMTS MUST be able to send SYSLOG event and (or) SNMP TRAP to the NMS.

---

2. Revised this event statement per ECN 0SS2-N-03046.

1. Replaced text and tables from this paragraph, through Section 7.4.3 per ECN OSS2-N-02193 by GO on 02/11/03.

2. Revised Table 7-2 and added two table footnotes per ECN OSS2-N-03034 and 0SS2-N-03046 by GO on 04/21/03 and 05/01/03.

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

Table 7-3 Default Event priorities for the 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 7-4 Default Event priorities for the 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.[1]

Table 7-5 Default Event priorities for the CMTS supporting both local-log non-volatile and 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	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

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

7.4.2.5 Event Priorities for DOCSIS and Vendor Specific Events[2]

DOCSIS 2.0 compliant cable device (CMTS/CM) MUST strictly assign DOCSIS and Vendor specific events accordingly to Table 7-6.[3]

Table 7-6 Event Priorities Assignment For CMs and CMTSes

Event Priority	CM Event Assignment	CMTS Event Assignment
1 Emergency	Vendor Specific	Vendor Specific
2 Alert	DOCSIS	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

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

1. Revised Table 7-5 and added two table footnotes per ECN OSS2-N-03034 by GO on 04/21/03.
2. Added this section per ECN OSS2-N-02193 by GO on 02/11/03.
3. Revised Table 7-6 per ECN OSS2-N-03046 by GO on 05/01/03.

7.4.3 Throttling, Limiting and Priority for Event, Trap and Syslog

7.4.3.1 Trap and Syslog Throttling, Trap and Syslog Limiting

DOCSIS 2.0-compliant cable devices (CMTS/CM) MUST support SNMP TRAP/INFORM and SYSLOG throttling and limiting as described in DOCS-CABLE-DEVICE-MIB [RFC 2669], regardless of SNMP mode.

7.4.3.2 Maximum Priorities for Event Reporting

Table 7-2 and Table 7-3, Table 7-4, and Table 7-5 in Section 7.4.2 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 these tables. According to the priority definitions, there is a maximum level that an event can be reported. Table 7-7 shows that maximum level for CM events and Table 7-8 displays that for CMTS events.

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

Table 7-7 Maximum Level of Support for CM 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	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 7-8 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				

7.4.3.3 BIT Values for docsDevEvReporting [RFC 2669]

Permissible BIT values for [RFC 2669] docsDevEvReporting objects 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 values.

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.

7.4.4 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.

7.5 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]. 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.

7.5.1 Additional MIB implementation requirements

To support performance monitoring and data collection for capacity, fault, and performance management, CMs and CMTSes 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

7.6 Coexistence

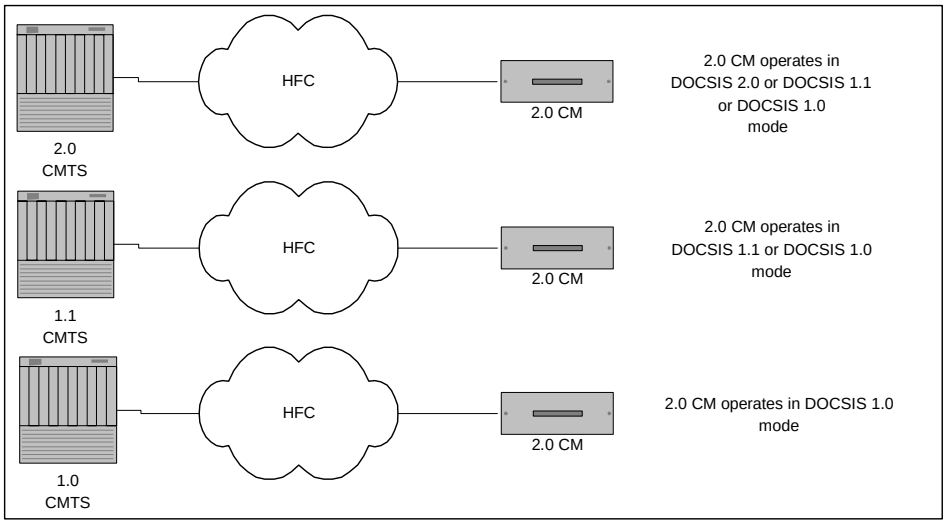


Figure 7-7 Coexistence (DOCSIS 1.0 mode vs. DOCSIS 1.1 mode vs. DOCSIS 2.0 mode)

When a DOCSIS 2.0-compliant CM is connected to a 2.0 CMTS, it can operate in DOCSIS 2.0, 1.1, or 1.0 mode.

When a DOCSIS 2.0-compliant CM is connected to a 1.1 CMTS, it can operate in either DOCSIS 1.1 or 1.0 mode.

When a DOCSIS 2.0-compliant CM is connected to a 1.0 CMTS, it operates in DOCSIS 1.0 mode.

Refer to [DOCSIS 5] and the BPI+ specification for more detailed descriptions of features available for DOCSIS 2.0-compliant CM operating modes.

7.6.1 Coexistence and MIBs

The following table summarizes the requirements for MIB support for a DOCSIS 2.0 CM operating in DOCSIS 2.0, 1.1, or 1.0 mode.

The table also addresses the cases where different sections of a MIB have different support requirements across CM operational modes.[1]

1. Revised Table 7-9 per ECN OSS2-N-03069 by GO on 07/11/03.

Table 7-9 DOCSIS 2.0 CM Modes and MIB Requirements

SNMP version:	v1/v2c RO	v1/v2c	v1/v2c/v3	v1/v2c	v1/v2c/v3	v1/v2c	v1/v2c/v3
2.0 CM	**Before Registration (RO from CMCI only)**	**Docsis 1.0 Mode + NmAccess**	**Docsis 1.0 Mode + SNMP Coexistence**	**Docsis 1.1 Mode + NmAccess**	**Docsis 1.1 Mode + SNMP Coexistence**	**Docsis 2.0 Mode + NmAccess**	**Docsis 2.0 Mode + SNMP Coexistence**
Access-ible MIBs	DOCS-IF-MIB	DOCS-IF-MIB	DOCS-IF-MIB	DOCS-IF-MIB	DOCS-IF-MIB	DOCS-IF-MIB	DOCS-IF-MIB
	IF-EXT (9)	RFC2863	RFC2863	IF-EXT (9)	IF-EXT (9)	RFC2863	RFC2863
	RFC2863	RFC1493	RFC1493	RFC2863	RFC2863	RFC1493	RFC1493
	RFC1493	RFC2669	RFC2669(1)	RFC1493	RFC1493	RFC2669	RFC2669
	RFC2669(1)	RFC2011	RFC2011	RFC2669	RFC2669(1)	RFC2011	(1)
	RFC2011(1)	RFC2013	RFC2013	RFC2011	RFC2011	RFC2013	RFC2011
	RFC2013	RFC3418	RFC3418	RFC2013	RFC2013	RFC3418	RFC2013
	RFC3418	RFC2665	RFC2665	RFC3418	RFC3418	RFC2665	RFC3418
	RFC2665	RFC2933 (7)	RFC2933 (7)	RFC2665	RFC2665	RFC2933	RFC2665
	RFC2933 (1, 7)	RFC3083	RFC3083	RFC2933	RFC2933	QOS	RFC2933
	RFC3083	BPI+ (6)	BPI+ (6)	QOS	QOS	BPI+	QOS
	QOS	USB	USB	BPI+	BPI+	USB	BPI+
	QOS	TRAP(2)	RFC2786	USB	RFC2786	TRAP	RFC2786
	BPI+		RFC3411	TRAP	RFC3411		RFC3411
	USB		RFC3412		RFC3412		RFC3412
	TRAP		RFC3413		RFC3413		RFC3413
			RFC3414		RFC3414		RFC3414
			RFC3415		RFC3415		RFC3415
			RFC2576		RFC2576		RFC2576
			TRAP(2)		USB		USB
					TRAP		TRAP

Table 7-9 DOCSIS 2.0 CM Modes and MIB Requirements (Continued)

SNMP version:	v1/v2c RO	v1/v2c	v1/v2c/v3	v1/v2c	v1/v2c/v3	v1/v2c	v1/v2c/v3
2.0 CM	**Before Registration (RO from CMCI only)**	**Docsis 1.0 Mode + NmAccess**	**Docsis 1.0 Mode + SNMP Coexistence**	**Docsis 1.1 Mode + NmAccess**	**Docsis 1.1 Mode + SNMP Coexistence**	**Docsis 2.0 Mode + NmAccess**	**Docsis 2.0 Mode + SNMP Coexistence**
Inaccess-ible MIBs	RFC2669	QOS(5)	RFC2669	RFC2786 (4)	RFC2669	IF-EXT(9)	IF-EXT(9)
	NmAccessTable	RFC2786 (4)	NmAccessTable(3)	RFC3411 (5)	NmAccessTable(3)	RFC2786 (4)	RFC2669
	RFC2669	RFC3411	QOS(5)	RFC3412	RFC3083	RFC3411(5)	NmAccessTable
	CpeTable	RFC3412(5)		RFC3413 (5)		RFC3412(5)	RFC3083
	RFC2011	RFC3413		RFC3414		RFC3413	
	ipAddrTable (8)	RFC3414 (5)		RFC3415 (5)		RFC3414 (5)R	
	RFC2011	RFC3415 (5)		RFC2576 (5)		RFC3415(5)	
	ipNetToMedia(8)						
	RFC2933						
	InterfaceQuerier(8)	RFC2576 (5)		RFC3083 (5)		RFC2576 (5)	
	RFC2933					RFC3083(5)	
	IGMP						
	CacheTable (8)						
	RFC2786						
	RFC3411						
	RFC3412						
	RFC3413						
	RFC3414						
	RFC3415						
	RFC2576						

Notes to Table 7-9:

1. Part of a mib is not accessible. See Inaccessible section for inaccessible objects.
2. Supporting this MIB is optional. When DOCSIS 2.0 CM operates at 1.0 mode, it MAY (optionally) support DOCS-CABLE-DEVICE-TRAP-MIB. Some of the traps will not be applicable. Please see Appendix A for details.
3. RFC 2669 - When CM is in SNMP Coexistence mode, the CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all requests to tables and objects in docsDevNmAccessTable.
4. RFC 2786 - When CM is in SNMP V1/V2c NmAccess mode, CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all requests to tables and objects in this MIB.
5. When CM is in SNMP v1/v2c NmAccess mode, CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all requests to tables and objects defined.
6. BPI+ MIB Part of DOCS-BPI2-MIB MUST be supported to enable secure software download. Refer to Appendix A for specificMIB object requirements. For all other objects in DOCS-BPI2-MIB CM MUST respond with "NoSuchName" or correspondingSNMPv2c error code "NoAccess" for all requests.
7. Supporting this MIB is optional. If CM in 1.0 mode supports IGMP, it must implement RFC 2933.
8. Access to this object SHOULD be prohibited as it contains IP address object(s). Refer to Section 5.2 for more details.

9. This MIB has been deprecated and MUST NOT be supported in DOCSIS 2.0 modes.
10. This MIB has been deprecated and is optional in DOCSIS 1.1 modes.

The following MIB Identities are used in Table 7-9:[1]

[RFC 1493] - BRIDGE-MIB
[RFC 3418] - SNMPv2-MIB
[RFC 2011] - IP-MIB
[RFC 2013] - UDP
[RFC 3411] - SNMP-FRAMEWORK-MIB
[RFC 3412] - SNMP-MPD-MIB
[RFC 3413] - SNMP Applications: SNMP-TARGET-MIB, SNMP-NOTIFICATION-MIB
[RFC 3414] - SNMP-USER-BASED-SM-MIB
[RFC 3415] - SNMP-VIEW-BASED-ACM-MIB
[RFC 2576] - SNMP-COMMUNITY-MIB
[RFC 2665] - EtherLike-MIB
[RFC 2669] - DOCS-CABLE-DEVICE-MIB
[RFC 2786] - SNMP-USM-DH-OBJECTS-MIB
[RFC 2863] - IF-MIB
[RFC 2933] - IGMP-STD-MIB
[RFC 3083] - DOCS-BPI-MIB
BPI+ - DOCS-BPI2-MIB
IF-EXT - DOCS-IF-EXT-MIB (see Annex G)
QOS - DOCS-QOS-MIB
DOCS-IF-MIB - Supersedes RFC 2670 draft for DOCSIS 2.0 DOCSIS device.
Trap - DOCS-CABLE-DEVICE-TRAP-MIB (See Annex H)
USB - USB-MIB

7.6.2 Coexistence and SNMP

A DOCSIS 2.0-compliant CM MUST support SNMPv3 and SNMPv1/v2c functionality as specified in Section 5 regardless of what mode (DOCSIS 1.0, 1.1, or 2.0) the CM operates in.

1. Revised MIB Identities per ECN OSSI2-N-03067 by GO on 02/17/04.

8 OSSI for BPI+

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

8.1 DOCSIS Root CA

The DOCSIS Root CA issues two kinds of digital certificates as specified by the BPI+ specification. One is the Manufacturer CA Certificate embedded in the DOCSIS 2.0-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 Secure Software Downloading regardless of whether the BPI+ is provisioned or not.

The legitimate DOCSIS Root CA Certificate needs to be delivered to cable operators and/or 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 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 cable operators, CMTS vendors, and CM vendors.

8.2 Digital Certificate Validity Period and Re-issuance

8.2.1 DOCSIS Root CA Certificate

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

8.2.2 DOCSIS Manufacturer CA Certificate

When the DOCSIS Root CA newly issues the DOCSIS Manufacturer CA Certificate, the following conditions apply:

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

Before the DOCSIS Manufacturer CA Certificate expires, a certificate with the same information except the tbsCertificate.validity.notAfter and tbsCertificate.serialNumber needs to be re-issued. DOCSIS 2.0-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 as the current DOCSIS Manufacturer CA Certificate:

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

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

8.2.3 DOCSIS CM Certificate

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

8.2.4 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 value MUST be the actual issuance date and time
- tbsCertificate.validity.notAfter MUST NOT exceed an actual issuance date and time by 10 years, and MUST be valid at least 2 years from the actual issuance date.

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. DOCSIS 2.0-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[1]

As well, the tbsCertificate.validity.notBefore value MUST be between the tbsCertificate.validity.notBefore value of the current DOCSIS Manufacturer CVC, and the actual issuance date and time. (That is, the tbsCertificate.validity.notBefore value can be the same as the tbsCertificate.validity.notBefore value of the current DOCSIS Manufacturer CVC, the actual issuance date and time, or any value between there two values.) In addition, the tbsCertificate.validity.notAfter MUST be the actual re-issuance date and time plus 2 to 10 years.[2]

8.3 CM Code File Signing Policy

CM vendors and cable operators can control the Secure Software Download process based on their policies by updating the Manufacturer/Co-Signer CVC or by changing the signingTime in the Manufacturer/Co-Signer CVS (Code Verification Signature). At this time, the DOCSIS 2.0 specifications do not specify the policy related to the CM Code File signing process. However, an example of the policy is specified in this section.

8.3.1 Manufacturer CM Code File Signing Policy

A DOCSIS 2.0-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 exactly the 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.

Once the vendor realizes there are certain issues in one or more CM Code File(s) and wants to revoke them, the vendor chooses 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, it re-signs all the still-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 CM's software can be downgraded if necessary.

---

1. Revised preceding paragraph and bullet statements per ECN OSS2-N-03053 by GO on 06/04/03.

2. Revised this paragraph per ECN OSSIv2.0-N-0130-2 by GO on 3/17/04.

9 OSSI for CMCI

This section defines the operational mechanisms needed to support the transmission of data over cable services between a cable modem and customer premise equipment. More specifically, this section outlines 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, a Universal Serial Bus (USB) CMCI interface, or both. If both interfaces are present on a CM, they MAY be active at the same time.

Internal cable modems MUST utilize the Peripheral Component Interconnect (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.

A CPE-Controlled Cable modem's (CCCM) CMCI MAY be either a Peripheral Component Interconnect (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.

9.1 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 5.2. 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 drop by CM.

CM MAY also implement alternative interfaces like link-local method described in the IETF document "draft-ietf-zeroconf-ipv4-linklocal-10.txt" [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.[1]

9.2 Console Access

An external cable modem MUST NOT allow access to the CM functions via a console port. In 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 2.0-defined RF interfaces and operator-controlled SNMP access via the CMCI.

---

1. Revised the first sentence of this paragraph per ECN OSSIv2.0-N-03.0117-2 by GO on 2/19/04.

9.3 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 5.2 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.

9.4 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 7.3. 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].

9.5 Management Information Base (MIB) Requirements

All Cable Modems MUST implement the MIBs detailed in Section 6 of this specification, with the following exceptions:

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

10 CM Operational Status Visualization[1]

DOCSIS 2.0 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.

10.1 CM LEDs Requirements and Operation

The LEDs on a DOCSIS 2.0 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 2.0 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, and ONLINE
- CPE: This Group has the LINK LED indication. It SHOULD have a minimum of 1 LED labeled as LINK. DOCSIS 2.0 CMs MAY have multiple LEDs in the CPE Group to represent individual CPE interfaces types and parameters. Those LEDs MAY be labeled according 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, and ONLINE, from left to right, or Top to Bottom, as appropriate for the orientation of the device. As well, the overall LED distribution SHOULD intend to be in the order POWER, DS, US, ONLINE, and 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

Note: RECOMMENDED LEDs SHOULD operate as described below:

10.1.1 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

1. Added Section 10 per ECN OSS2-N-03025 by GO on 06/13/03.

is properly connected (see Section 10.1.5 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'.

10.1.2 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'.

10.1.3 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 in [DOCSIS 5]).

10.1.4 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 [DOCSIS 5], 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'.

10.1.5 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 the power and self test step is completed.

10.2 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 2.0 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 10.1.4 and 10.1.5).

10.2.1 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'.

Annex A Detailed MIB Requirements (normative)[1]

The following abbreviations and rules apply in this Annex:

ACC-FN Accessible for Notify.

ATRAP Accessible through SNMP trap.

D Deprecated. 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 instantiate 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. The device MUST NOT support the object. That is, an agent MUST NOT instantiate 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 instantiate 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 2.0 OSSI lets vendors choose whether or not to support the obsolete object. 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, the SNMP agent MUST NOT instantiate 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.

1. Revised following table per ECN OSS2-N-03021 (rescinded OSS-N-02234), OSS2-N-03014, OSS2-N-03023, OSS2-N-03069, OSS2-N-03071, OSS2-N-03067, OSS2-N-03090, and OSSIv2.0-N-04.0127-4 by GO on 02/26/03, 03/2/03, 07/11/03, 11/17/03, and 3/16/04.

DOCS-IF-MIB (DOCS-IF-MIB: draft-ietf-ipcdn-docs-rfmibv2-05.txt)								
docsIfDownstreamChannelTable								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsIfDownChannelId	M	RO	M	RO	M	RO	M	RO
docsIfDownChannelFrequency	M	RO	M	RO	M	RO	M	RW/RO
docsIfDownChannelWidth	M	RO	M	RO	M	RO	M	RW/RO
docsIfDownChannelModulation	M	RO	M	RO	M	RO	M	RW
docsIfDownChannelInterleave	M	RO	M	RO	M	RO	M	RW
docsIfDownChannelPower	M	RO	M	RO	M	RO	M	RW/RO
docsIfDownChannelAnnex	O	RO	O	RO	M	RO	M	RW/RO
docsIfUpstreamChannelTable								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsIfUpChannelId	M	RO	M	RO	M	RO	M	RO
docsIfUpChannelFrequency	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelWidth	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelModulationProfile	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelSlotSize	M	RO	M	RO	M	RO	M	RC/RO
docsIfUpChannelTxTimingOffset	M	RO	M	RO	M	RO	M	RO
docsIfUpChannelRangingBackoffStart	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelRangingBackoffEnd	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelTxBackoffStart	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelTxBackoffEnd	M	RO	M	RO	M	RO	M	RC
docsIfUpChannelScdmaActiveCodes	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelScdmaCodesPerSlot	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelScdmaFrameSize	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelScdmaHoppingSeed	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelType	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelCloneFrom	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelUpdate	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelStatus	O	RO	O	RO	M	RO	M	RC
docsIfUpChannelPreEqEnable	O	RO	M	RO	M	RO	M	RC

docsIfQosProfileTable

Object	2.0 CMin 1.0 mode	Access	2.0 CMin 1.1 mode	Access	2.0 CMin 2.0 mode	Access	CMTS	Access
docsIfQosProfIndex	M	N-Acc	O	N-Acc	O	N-Acc	O	N-Acc
docsIfQosProfPriority	M	RO	O	RO	O	RO	O	RC/RO
docsIfQosProfMaxUpBandwidth	M	RO	O	RO	O	RO	O	RC/RO
docsIfQosProfGuarUpBandwidth	M	RO	O	RO	O	RO	O	RC/RO
docsIfQosProfMaxDownBandwidth	M	RO	O	RO	O	RO	O	RC/RO
docsIfQosProfMaxTxBurst	D	RO	D	RO	D	RO	D	RC/RO
docsIfQosProfBaselinePrivacy	M	RO	O	RO	O	RO	O	RC/RO
docsIfQosProfStatus	M	RO	O	RO	O	RO	O	RC/RO
docsIfQosProfMaxTransmitBurst	M	RO	O	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	2.0 CMin 1.0 mode	Access	2.0 CMin 1.1 mode	Access	2.0 CMin 2.0 mode	Access	CMTS	Access
docsIfSigQExtUnerroreds	O	RO	O	RO	M	RO	M	RO
docsIfSigQExtCorrecteds	O	RO	O	RO	M	RO	M	RO
docsIfSigQExtUncorrectables	O	RO	O	RO	M	RO	M	RO

(no table)

Object	2.0 CMin 1.0 mode	Access	2.0 CMin 1.1 mode	Access	2.0 CMin 2.0 mode	Access	CMTS	Access
docsIfDocsisBaseCapability	O	RO	M	RO	M	RO	M	RO

docsIfCmMacTable

Object	CM	Access	CMTS	Access
docsIfCmCmtsAddress	M	RO	NA	NA
docsIfCmCapabilities	M	RO	NA	NA

docsIfCmRangingTimeout					Ob	N-Sup	NA	NA
docsIfCmRangingTimeout					M	RW	NA	NA
docsIfCmStatusTable								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsIfCmStatusValue	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusCode	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusTxPower	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusResets	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusLostSyncs	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusInvalidMaps	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusInvalidUcds	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusInvalidRanging Responses	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusInvalidRegistration Responses	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusT1Timeouts	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusT2Timeouts	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusT3Timeouts	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusT4Timeouts	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusRangingAborteds	M	RO	M	RO	M	RO	NA	NA
docsIfCmStatusDocsisOperMode	O	RO	M	RO	M	RO	NA	NA
docsIfCmStatusModulationType	O	RO	M	RO	M	RO	NA	NA
docsIfCmStatusEqualizationData	O	RO	M	RO	M	RO	NA	NA
(no table)								
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
docsIfCmtsUpChnlCtrUcastGrantedMslot	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
docsIfCmtsUpChnlCtrExtCollCntnInitMaintMslots	NA	NA	O	RO

docsIfCmServiceTable

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsIfCmServiceId	M	N-Acc	M	N-Acc	M	N-Acc	NA	NA
docsIfCmServiceQosProfile	M	RO	M	RO	M	RO	NA	NA
docsIfCmServiceTxSlotsImmed	M	RO	M	RO	M	RO	NA	NA
docsIfCmServiceTxSlotsDed	M	RO	M	RO	M	RO	NA	NA
docsIfCmServiceTxRetries	M	RO	M	RO	M	RO	NA	NA

docsIfCmServiceTxExceeds	M	RO	M	RO	M	RO	NA	NA
docsIfCmServiceRqRetries	M	RO	M	RO	M	RO	NA	NA
docsIfCmServiceRqExceeds	M	RO	M	RO	M	RO	NA	NA
docsIfCmServiceExtTxSlotsImmed	O	RO	O	RO	M	RO	NA	NA
docsIfCmServiceExtTxSlotsDed	O	RO	O	RO	M	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
docsIfCmtsInsertionInterval	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	D	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	M	RO

docsIfCmtsCmStatusExtCorrecteds	NA	NA	M	RO
docsIfCmtsCmStatusExtUncorrectables	NA	NA	M	RO
docsIfCmtsCmStatusDocsisRegMode	NA	NA	M	RO
docsIfCmtsCmStatusModulationType	NA	NA	M	RO
docsIfCmtsCmStatusInetAddressType	NA	NA	M	RO
docsIfCmtsCmStatusInetAddress	NA	NA	M	RO
docIfCmtsCmStatusValueLastUpdate	NA	NA	M	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	M	RC
docsIfCmtsModByteInterleaverBlockSize	NA	NA	M	RC
docsIfCmtsModPreambleType	NA	NA	M	RC
docsIfCmtsModTcmErrorCorrectionOn	NA	NA	M	RC
docsIfCmtsModScdmaInterleaverStepSize	NA	NA	M	RC
docsIfCmtsModScdmaSpreaderEnable	NA	NA	M	RO
docsIfCmtsModScdmaSubframeCodes	NA	NA	M	RC
docsIfCmtsModChannelType	NA	NA	M	RC

Object	CM	Access	CMTS	Access
docsIfCmtsQosProfilePermissions	NA	NA	M	RW / RO

docsIfCmtsMacToCmTable

Object	CM	Access	CMTS	Access
docsIfCmtsCmMac	NA	NA	M	N-Acc
docsIfCmtsCmPtr	NA	NA	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	M	RC/RO	M	RC/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	CM	Access	CMTS	Access
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 and DOCSIS 2.0; 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: CMs supporting the IP spoofing function MUST implement this group. CMs not supporting the 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
Object	**CM**	**Access**	**CMTS**	**Access**
ipRoutingDiscards	M	RO	M	RO
ICMP Group				
Objects	**CM**	**Access**	**CMTS**	**Access**
icmpInMsgs	M	RO	M	RO
icmpInErrors	M	RO	M	RO
icmpInDestUnreachs	M	RO	M	RO
icmpInTimeExcds	M	RO	M	RO
icmpInParmProbs	M	RO	M	RO
icmpInSrcQuenchs	M	RO	M	RO
icmpInRedirects	M	RO	M	RO
icmpInEchos	M	RO	M	RO
icmpInEchosReps	M	RO	M	RO
icmpInTimestamps	M	RO	M	RO
icmpInTimeStampreps	M	RO	M	RO
icmpInAddrMasks	M	RO	M	RO
icmpInAddrMaskReps	M	RO	M	RO
icmpOutMsgs	M	RO	M	RO
icmpOutErrors	M	RO	M	RO
icmpOutDestUnreachs	M	RO	M	RO
icmpOutTimeExcds	M	RO	M	RO
icmpOutParmProbs	M	RO	M	RO
icmpOutSrcQuenchs	M	RO	M	RO
icmpOutRedirects	M	RO	M	RO
icmpOutEchos	M	RO	M	RO
icmpOutEchoReps	M	RO	M	RO
icmpOutTimestamps	M	RO	M	RO
icmpOutTimestampReps	M	RO	M	RO
icmpOutAddrMasks	M	RO	M	RO
icmpOutAddrMaskReps	M	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
snmpSet Group				
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 CMs that support 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 (Annex J)

NOTE: 2.0 CMs in 1.0 mode MUST NOT support this MIB.

docsQosPktClassTable

Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosPktClassId	M	N-Acc	M	N-Acc	M	N-Acc
docsQosPktClassDirection	M	RO	M	RO	M	RO
docsQosPktClassPriority	M	RO	M	RO	M	RO
docsQosPktClassIpTosLow	M	RO	M	RO	M	RO
docsQosPktClassIpTosHigh	M	RO	M	RO	M	RO
docsQosPktClassIpTosMask	M	RO	M	RO	M	RO
docsQosPktClassIpProtocol	M	RO	M	RO	M	RO
docsQosPktClassIpSourceAddr	M	RO	M	RO	M	RO
docsQosPktClassIpSourceMask	M	RO	M	RO	M	RO
docsQosPktClassIpDestAddr	M	RO	M	RO	M	RO

docsQosPktClassIpDestMask	M	RO	M	RO	M	RO
docsQosPktClassSourcePortStart	M	RO	M	RO	M	RO
docsQosPktClassSourcePortEnd	M	RO	M	RO	M	RO
docsQosPktClassDestPortStart	M	RO	M	RO	M	RO
docsQosPktClassDestPortEnd	M	RO	M	RO	M	RO
docsQosPktClassDestMacAddr	M	RO	M	RO	M	RO
docsQosPktClassDestMacMask	M	RO	M	RO	M	RO
docsQosPktClassSourceMacAddr	M	RO	M	RO	M	RO
docsQosPktClassEnetProtocolType	M	RO	M	RO	M	RO
docsQosPktClassEnetProtocol	M	RO	M	RO	M	RO
docsQosPktClassUserPriLow	M	RO	M	RO	M	RO
docsQosPktClassUserPriHigh	M	RO	M	RO	M	RO
docsQosPktClassVlanId	M	RO	M	RO	M	RO
docsQosPktClassState	M	RO	M	RO	M	RO
docsQosPktClassPkts	M	RO	M	RO	M	RO
docsQosPktClasBitMap	M	RO	M	RO	M	RO

docsQosParamSetTable

Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosParamSetServiceClassName	M	RO	M	RO	M	RO
docsQosParamSetPriority	M	RO	M	RO	M	RO
docsQosParamSetMaxTrafficRate	M	RO	M	RO	M	RO
docsQosParamSetMaxTrafficBurst	M	RO	M	RO	M	RO
docsQosParamSetMinReservedRate	M	RO	M	RO	M	RO
docsQosParamSetMinReservedPkt	M	RO	M	RO	M	RO
docsQosParamSetActiveTimeout	M	RO	M	RO	M	RO
docsQosParamSetAdmittedTimeout	M	RO	M	RO	M	RO
docsQosParamSetMaxConcatBurst	M	RO	M	RO	M	RO
docsQosParamSetSchedulingType	M	RO	M	RO	M	RO
docsQosParamSetNomPollInterval	M	RO	M	RO	M	RO
docsQosParamSetTolPollJitter	M	RO	M	RO	M	RO
docsQosParamSetUnsolicitGrantSize	M	RO	M	RO	M	RO
docsQosParamSetNomGrantInterval	M	RO	M	RO	M	RO
docsQosParamSetTolGrantJitter	M	RO	M	RO	M	RO
docsQosParamSetGrantsPerInterval	M	RO	M	RO	M	RO
docsQosParamSetTosAndMask	M	RO	M	RO	M	RO
docsQosParamSetTosOrMask	M	RO	M	RO	M	RO
docsQosParamSetMaxLatency	M	RO	M	RO	M	RO
docsQosParamSetType	M	NA	M	NA	M	NA
docsQosParamSetRequestPolicyOct	M	RO	M	RO	M	RO
docsQosParamSetBitMap	M	RO	M	RO	M	RO

docsQosServiceFlowTable

Object	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsQosServiceFlowId	M	N-Acc	M	N-Acc	M	N-Acc
docsQosServiceFlowSID	M	RO	M	RO	M	RO
docsQosServiceFlowDirection	M	RO	M	RO	M	RO
docsQosServiceFlowPrimary	M	RO	M	RO	M	RO

docsQosServiceFlowStatsTable

Object	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsQosServiceFlowPkts	M	RO	M	RO	M	RO
docsQosServiceFlowOctets	M	RO	M	RO	M	RO
docsQosServiceFlowTimeCreated	M	RO	M	RO	M	RO
docsQosServiceFlowTimeActive	M	RO	M	RO	M	RO
docsQosServiceFlowPHSUnknowns	M	RO	M	RO	M	RO
docsQosServiceFlowPolicedDropPkts	M	RO	M	RO	M	RO
docsQosServiceFlowPolicedDelayPkts	M	RO	M	RO	M	RO

docsQosUpstreamStatsTable

Object	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsQosSID	N-Sup		N-Sup		M	N-Acc
docsQosUpstreamFragments	N-Sup		N-Sup		M	RO
docsQosUpstreamFragDiscards	N-Sup		N-Sup		M	RO
docsQosUpstreamConcatBursts	N-Sup		N-Sup		M	RO

docsQosDynamicServiceStatsTable

Object	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsQosIfDirection	M	N-Acc	M	N-Acc	M	N-Acc
docsQosDSAReqs	M	RO	M	RO	M	RO
docsQosDSARsps	M	RO	M	RO	M	RO
docsQosDSAAcks	M	RO	M	RO	M	RO

docsQosDSCReqs	M	RO	M	RO	M	RO
docsQosDSCRsps	M	RO	M	RO	M	RO
docsQosDSCAcks	M	RO	M	RO	M	RO
docsQosDSDReqs	M	RO	M	RO	M	RO
docsQosDSDRsps	M	RO	M	RO	M	RO
docsQosDynamicAdds	M	RO	M	RO	M	RO
docsQosDynamicAddFails	M	RO	M	RO	M	RO
docsQosDynamicChanges	M	RO	M	RO	M	RO
docsQosDynamicChangeFails	M	RO	M	RO	M	RO
docsQosDynamicDeletes	M	RO	M	RO	M	RO
docsQosDynamicDeleteFails	M	RO	M	RO	M	RO
docsQosDCCReqs	M	RO	M	RO	M	RO
docsQosDCCRsps	M	RO	M	RO	M	RO
docsQosDCCAcks	M	RO	M	RO	M	RO
docsQosDCCs	M	RO	M	RO	M	RO
docsQosDCCFails	M	RO	M	RO	M	RO
DocsQosDCCRspDeparts	M	RO	M	RO	M	RO
DocsQosDCCRspArrives	M	RO	M	RO	M	RO

docsQosServiceFlowLogTable

Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosServiceFlowLogIndex	N-Sup		N-Sup		M	N-Acc
docsQosServiceFlowLogIfIndex	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogSFID	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogCmMac	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogPkts	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogOctets	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogTimeDeleted	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogTimeCreated	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogTimeActive	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogDirection	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogPrimary	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogServiceClassName	N-Sup		N-Sup		M	RO

docsQosServiceFlowLogPolicedDropPkts	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogPolicedDelayPkts	N-Sup		N-Sup		M	RO
docsQosServiceFlowLogControl	N-Sup		N-Sup		M	RW
docsQosServiceClassTable						
Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosServiceClassName	N-Sup		N-Sup		M	N-Acc
docsQosServiceClassStatus	N-Sup		N-Sup		M	RC
docsQosServiceClassPriority	N-Sup		N-Sup		M	RC
docsQosServiceClassMaxTrafficRate	N-Sup		N-Sup		M	RC
docsQosServiceClassMaxTrafficBurst	N-Sup		N-Sup		M	RC
docsQosServiceClassMinReservedRate	N-Sup		N-Sup		M	RC
docsQosServiceClassMinReservedPkt	N-Sup		N-Sup		M	RC
docsQosServiceClassMaxConcatBurst	N-Sup		N-Sup		M	RC
docsQosServiceClassNomPollInterval	N-Sup		N-Sup		M	RC
docsQosServiceClassTolPollJitter	N-Sup		N-Sup		M	RC
docsQosServiceClassUnsolicitGrantSize	N-Sup		N-Sup		M	RC
docsQosServiceClassNomGrantInterval	N-Sup		N-Sup		M	RC
docsQosServiceClassTolGrantJitter	N-Sup		N-Sup		M	RC
docsQosServiceClassGrantsPerInterval	N-Sup		N-Sup		M	RC
docsQosServiceClassMaxLatency	N-Sup		N-Sup		M	RC
docsQosServiceClassActiveTimeout	N-Sup		N-Sup		M	RC
docsQosServiceClassAdmittedTimeout	N-Sup		N-Sup		M	RC
docsQosServiceClassSchedulingTime	N-Sup		N-Sup		M	RC
docsQosServiceClassRequestPolicy	N-Sup		N-Sup		M	RC

docsQosServiceClassTosAndMask	N-Sup		N-Sup		M	RC
docsQosServiceClassTosOrMask	N-Sup		N-Sup		M	RC
docsQosServiceClassDirection	N-Sup		N-Sup		M	RC
docsQosServiceClassPolicyTable						
Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosServiceClassPolicyIndex	O	N-Acc	O	N-Acc	O	N-Acc
docsQosServiceClassPolicyName	O	RC	O	RC	O	RC
docsQosServiceClassPolicyRulePriority	O	RC	O	RC	O	RC
docsQosServiceClassPolicyStatus	O	RC	O	RC	O	RC
docsQosPHSTable						
Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosPHSField	M	RO	M	RO	O	RO
docsQosPHSMask	M	RO	M	RO	O	RO
docsQosPHSSize	M	RO	M	RO	O	RO
docsQosPHSVerify	M	RO	M	RO	O	RO
docsQosPHSIndex	M	RO	M	RO	O	RO
docsQosCmtsMacToSrvFlowTable						
Object	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsQosCmtsCmMac	N-Sup		N-Sup		M	N-Acc
docsQosCmtsServiceFlowId	N-Sup		N-Sup		M	N-Acc
docsQosCmtsIfIndex	N-Sup		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

Objects	**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 CMTSes.

NOTE: 2.0 CMs in 1.0 mode are not required to implement [RFC 2933].

IgmpInterfaceTable

Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
IgmpInterfaceIfIndex	O	N-Acc	M	N-Acc	M	N-Acc	M	N-Acc
igmpInterfaceQueryInterval	O	RC	M	RC	M	RC	M	RC
igmpInterfaceStatus	O	RC	M	RC	M	RC	M	RC
igmpInterfaceVersion	O	RC	M	RC	M	RC	M	RC
igmpInterfaceQuerier	O	RO	M	RO	M	RO	M	RO
igmpInterfaceQueryMaxResponseTime	O	RO	M	RO	M	RO	M	RO
igmpInterfaceVersion1QuerierTimer	O	RO	M	RO	M	RO	M	RO
igmpInterfaceWrongVersionQueries	O	RO	M	RO	M	RO	M	RO
igmpInterfaceJoins	O	RO	M	RO	M	RO	M	RO
IgmpInterfaceGroups	O	RO	M	RO	M	RO	M	RO
igmpInterfaceRobustness	O	RC	M	RC	M	RC	M	RC
igmpInterfaceLastMembQueryIntvl	O	RC	M	RC	M	RC	M	RC
igmpInterfaceProxyIfIndex	O	RC	M	RC	M	RC	M	RC
igmpInterfaceQuerierUpTime	O	RO	M	RO	M	RO	M	RO
igmpInterfaceQuerierExpiryTime	O	RO	M	RO	M	RO	M	RO

igmpCacheTable

Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
igmpCacheAddress	O	N-Acc	M	N-Acc	M	N-Acc	M	N-Acc
igmpCacheIfIndex	O	N-Acc	M	N-Acc	M	N-Acc	M	N-Acc
IgmpCacheSelf	O	RC	M	RC	M	RC	M	RC

igmpCacheLastReporter	O	RO	M	RO	M	RO	M	RO
igmpCacheUpTime	O	RO	M	RO	M	RO	M	RO
igmpCacheExpiryTime	O	RO	M	RO	M	RO	M	RO
igmpCacheStatus	O	RC	M	RC	M	RC	M	RC
igmpCacheVersion1HostTimer	O	RO	M	RO	M	RO	M	RO

Account Management MIB (MIB defining work is still in progress.)

docsCpeSegmentTable

Object	**CM**	**Access**	**CMTS**	**Access**
docsCpeSegmentID	NA	NA	O	RO
docsCpeSegmentIp	NA	NA	O	RC

docsCpeTrafficData Table

Objects	**CM**	**Access**	**CMTS**	**Access**
docsCpeIpAddress	NA	NA	O	RO
docsCpeTrafficDataUpStreamPackets	NA	NA	O	RC
docsCpeTrafficDataDownStreamPackets	NA	NA	O	RC
docsCpeTrafficDataUpStreamOctets	NA	NA	O	RC
docsCpeTrafficDataDownStreamOctets	NA	NA	O	RC
docsCpeTrafficDataUpStreamDropPackets	NA	NA	O	RC
docsCpeTrafficDataDownStreamDropPackets	NA	NA	O	RC

docsCmCpeTable	**CM**	**Access**	**CMTS**	**Access**
docsCmMacAddress	NA	NA	O	RC
docsCmIpAddress	NA	NA	O	RC
docsCpeMACAddress	NA	NA	O	RC
docsCpeIpAddress	NA	NA	O	RC

DOCS-BPI-MIB [RFC 3083]

docsBpiCmBaseTable

Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiCmPrivacyEnable	M	RO	N-Sup		N-Sup		NA	
docsBpiCmPublicKey	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthState	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthKeySequenceNumber	M	RO	N-Sup		N-Sup		NA	

docsBpiCmAuthExpires	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthReset	M	RW	N-Sup		N-Sup		NA	
docsBpiCmAuthGraceTime	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKGraceTime	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthWaitTimeout	M	RO	N-Sup		N-Sup		NA	
docsBpiCmReauthWaitTimeout	M	RO	N-Sup		N-Sup		NA	
docsBpiCmOpWaitTimeout	M	RO	N-Sup		N-Sup		NA	
docsBpiCmRekeyWaitTimeout	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthRejectWaitTimeout	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthRequests	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthReplies	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthRejects	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthInvalids	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthRejectErrorCode	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthRejectErrorString	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthInvalidErrorCode	M	RO	N-Sup		N-Sup		NA	
docsBpiCmAuthInvalidErrorString	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKTable								
Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiCmTEKPrivacyEnable	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKState	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKExpiresOld	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKExpiresNew	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKKeyRequests	M	RO	N-Sup		N-Sup		NA	

docsBpiCmTEKKeyReplies	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKKeyRejects	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKInvalids	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKAuthPends	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKKeyRejectErrorCode	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKKeyRejectErrorString	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKInvalidErrorCode	M	RO	N-Sup		N-Sup		NA	
docsBpiCmTEKInvalidErrorString	M	RO	N-Sup		N-Sup		NA	

docsBpiCmtsBaseTable

Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiCmtsDefaultAuthLifetime	NA		NA		NA		N-Sup	
docsBpiCmtsDefaultTEKLifetime	NA		NA		NA		N-Sup	
docsBpiCmtsDefaultAuthGraceTime	NA		NA		NA		N-Sup	
docsBpiCmtsDefaultTEKGraceTime	NA		NA		NA		N-Sup	
docsBpiCmtsAuthRequests	NA		NA		NA		N-Sup	
docsBpiCmtsAuthReplies	NA		NA		NA		N-Sup	
docsBpiCmtsAuthRejects	NA		NA		NA		N-Sup	
docsBpiCmtsAuthInvalids	NA		NA		NA		N-Sup	

docsBpiCmtsAuthTable

Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiCmtsAuthCmMacAddress	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmPublicKey	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmKeySequence-Number	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmExpires	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmLifetime	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmGraceTime	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmReset	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmRequests	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmReplies	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmRejects	NA		NA		NA		N-Sup	
docsBpiCmtsAuthCmInvalids	NA		NA		NA		N-Sup	

docsBpiCmtsAuthRejectErrorCode	NA		NA		NA		N-Sup	
docsBpiCmtsAuthRejectErrorString	NA		NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorCode	NA		NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorString	NA		NA		NA		N-Sup	
docsBpiCmtsTEKTable								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiCmtsTEKLifetime	NA		NA		NA		N-Sup	
docsBpiCmtsTEKGraceTime	NA		NA		NA		N-Sup	
docsBpiCmtsTEKExpiresOld	NA		NA		NA		N-Sup	
docsBpiCmtsTEKExpiresNew	NA		NA		NA		N-Sup	
docsBpiCmtsTEKReset	NA		NA		NA		N-Sup	
docsBpiCmtsKeyRequests	NA		NA		NA		N-Sup	
docsBpiCmtsKeyReplies	NA		NA		NA		N-Sup	
docsBpiCmtsKeyRejects	NA		NA		NA		N-Sup	
docsBpiCmtsTEKInvalids	NA		NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorCode	NA		NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorString	NA		NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorCode	NA		NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorString	NA		NA		NA		N-Sup	
docsBpiIpMulticastMapTable								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiIpMulticastAddress	NA		NA		NA		N-Sup	
docsBpiIpMulticastprefixLength	NA		NA		NA		N-Sup	
docsBpiIpMulticastServiceId	NA		NA		NA		N-Sup	
docsBpiIpMulticastMapControl	NA		NA		NA		N-Sup	
docsBpiMulticastAuthTable								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpiMulticastServiceId	NA		NA		NA		N-Sup	
docsBpiMulticastCmMacAddress	NA		NA		NA		N-Sup	
docsBpiMulticastAuthControl	NA		NA		NA		N-Sup	
BPI+ MIB (draft-ietf-ipcdn-bpiplus-mib-05.txt)								
docsBpi2CmBaseTable								

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsBpi2CmPrivacyEnable	O	RO	M	RO	M	RO	NA	
docsBpi2CmPublicKey	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthState	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthKeySequenceNumber	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthExpiresOld	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthExpiresNew	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthReset	O	RW	M	RW	M	RW	NA	
docsBpi2CmAuthGraceTime	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKGraceTime	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthWaitTimeout	O	RO	M	RO	M	RO	NA	
docsBpi2CmReauthWaitTimeout	O	RO	M	RO	M	RO	NA	
docsBpi2CmOpWaitTimeout	O	RO	M	RO	M	RO	NA	
docsBpi2CmRekeyWaitTimeout	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthRejectWaitTimeout	O	RO	M	RO	M	RO	NA	
docsBpi2CmSAMapWaitTimeout	O	RO	M	RO	M	RO	NA	
docsBpi2CmSAMapMaxRetries	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthentInfos	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthRequests	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthReplies	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthRejects	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthInvalids	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorCode	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorString	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorCode	O	RO	M	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorString	O	RO	M	RO	M	RO	NA	

docsBpi2CmTEKTable

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsBpi2CmTEKSAId	O	N-Acc	M	N-Acc	M	N-Acc	NA	
docsBpi2CmTEKSAType	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKDataEncryptAlg	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKDataAuthentAlg	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKState	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKKeySequenceNumber	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKExpiresOld	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKExpiresNew	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKKeyRequests	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKKeyReplies	O	RO	M	RO	M	RO	NA	

docsBpi2CmTEKKeyRejects	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKInvalids	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKAuthPends	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorCode	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorString	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorCode	O	RO	M	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorString	O	RO	M	RO	M	RO	NA	

docsBpi2CmIpMulticastMapTable

Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmIpMulticastIndex	O	N-Acc	M	N-Acc	M	N-Acc	NA	
docsBpi2CmIpMulticastAddressType	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastAddress	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAId	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapState	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRequests	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapReplies	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejects	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorCode	O	RO	M	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorString	O	RO	M	RO	M	RO	NA	

docsBpi2CmDeviceCertTable

Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmDeviceCmCert	M	RW/RO	M	RW/RO	M	RW/RO	NA	
docsBpi2CmDeviceManufCert	M	RO	M	RO	M	RO	NA	

docsBpi2CmCryptoSuiteTable

Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmCryptoSuiteIndex	M	N-Acc	M	N-Acc	M	N-Acc	NA	
docsBpi2CmCryptoSuiteDataEncryptAlg	M	RO	M	RO	M	RO	NA	
docsBpi2CmCryptoSuiteDataAuthentAlg	M	RO	M	RO	M	RO	NA	

docsBpi2CmtsBaseEntryTable

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsBpi2CmtsDefaultAuthLifetime	NA		NA		NA		M	RW
docsBpi2CmtsDefaultTEKLifetime	NA		NA		NA		M	RW
docsBpi2CmtsDefaultSelfSignedManuf CertTrust	NA		NA		NA		M	RW
docsBpi2CmtsCheckCertValidity Periods	NA		NA		NA		M	RW
docsBpi2CmtsAuthentInfos	NA		NA		NA		M	RO
docsBpi2CmtsAuthRequests	NA		NA		NA		M	RO
docsBpi2CmtsAuthReplies	NA		NA		NA		M	RO
docsBpi2CmtsAuthRejects	NA		NA		NA		M	RO
docsBpi2CmtsAuthInvalids	NA		NA		NA		M	RO
docsBpi2CmtsSAMapRequests	NA		NA		NA		M	RO
docsBpi2CmtsSAMapReplies	NA		NA		NA		M	RO
docsBpi2CmtsSAMapRejects	NA		NA		NA		M	RO

docsBpi2CmtsAuthEntryTable

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsBpi2CmtsAuthCmMacAddress	NA		NA		NA		M	N-Acc
docsBpi2CmtsAuthCmBpiVersion	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmPublicKey	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmKeySequence Number	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresOld	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresNew	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmLifetime	NA		NA		NA		M	RW
docsBpi2CmtsAuthCmGraceTime	NA		NA		NA		Ob	RO
docsBpi2CmtsAuthCmReset	NA		NA		NA		M	RW
docsBpi2CmtsAuthCmInfos	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmRequests	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmReplies	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmRejects	NA		NA		NA		M	RO
docsBpi2CmtsAuthCmInvalids	NA		NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorCode	NA		NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorString	NA		NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorCode	NA		NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorString	NA		NA		NA		M	RO
docsBpi2CmtsAuthPrimarySAId	NA		NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCertValid	NA		NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCert	NA		NA		NA		M	RO

docsBpi2CmtsTEKTable

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsBpi2CmtsTEKSAId	NA		NA		NA		M	N-Acc
docsBpi2CmtsTEKSAType	NA		NA		NA		M	RO
docsBpi2CmtsTEKDataEncryptAlg	NA		NA		NA		M	RO
docsBpi2CmtsTEKDataAuthentAlg	NA		NA		NA		M	RO
docsBpi2CmtsTEKLifetime	NA		NA		NA		M	RW
docsBpi2CmtsTEKGraceTime	NA		NA		NA		Ob	RO
docsBpi2CmtsTEKKeySequenceNumber	NA		NA		NA		M	RO
docsBpi2CmtsTEKExpiresOld	NA		NA		NA		M	RO
docsBpi2CmtsTEKExpiresNew	NA		NA		NA		M	RO
docsBpi2CmtsTEKReset	NA		NA		NA		M	RW
docsBpi2CmtsKeyRequests	NA		NA		NA		M	RO
docsBpi2CmtsKeyReplies	NA		NA		NA		M	RO
docsBpi2CmtsKeyRejects	NA		NA		NA		M	RO
docsBpi2CmtsTEKInvalids	NA		NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorCode	NA		NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorString	NA		NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorCode	NA		NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorString	NA		NA		NA		M	RO

docsBpi2CmtsIpMulticastMapTable

Object	2.0 CM in 1.0 mode	Access	2.0 CM in 1.1 mode	Access	2.0 CM in 2.0 mode	Access	CMTS	Access
docsBpi2CmtsIpMulticastIndex	NA		NA		NA		M	N-Acc
docsBpi2CmtsIpMulticastAddressType	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastAddress	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastMaskType	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastMask	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAId	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAType	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastDataEncryptAlg	NA		NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastDataAuthentAlg	NA		NA		NA		M	RC/RO

docsBpi2CmtsIpMulticastSAMapRequests	NA		NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapReplies	NA		NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejects	NA		NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorCode	NA		NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorString	NA		NA		NA		M	RO
docsBpi2CmtsIpMulticastMapControl	NA		NA		NA		M	RC/RO
docsBpi2CmtsMulticastAuthTable								
Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsMulticastAuthSAId	NA		NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthCmMacAddress	NA		NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthControl	NA		NA		NA		M	RC/RO
docsBpi2CmtsProvisionedCmCertTable								
Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsProvisionedCmCertMacAddress	NA		NA		NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertTrust	NA		NA		NA		M	RC
docsBpi2CmtsProvisionedCmCertSource	NA		NA		NA		M	RO
docsBpi2CmtsProvisionedCmCertStatus	NA		NA		NA		M	RC
docsBpi2CmtsProvisionedCmCert	NA		NA		NA		M	RC
docsBpi2CmtsCACertTable								
Object	**2.0 CMin 1.0 mode**	**Access**	**2.0 CMin 1.1 mode**	**Access**	**2.0 CMin 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CmtsCACertIndex	NA		NA		NA		M	N-Acc
docsBpi2CmtsCACertSubject	NA		NA		NA		M	RO
docsBpi2CmtsCACertIssuer	NA		NA		NA		M	RO
docsBpi2CmtsCACertSerialNumber	NA		NA		NA		M	RO
docsBpi2CmtsCACertTrust	NA		NA		NA		M	RC
docsBpi2CmtsCACertSource	NA		NA		NA		M	RO
docsBpi2CmtsCACertStatus	NA		NA		NA		M	RC
docsBpi2CmtsCACert	NA		NA		NA		M	RC

docsBpi2CmtsCACertThumprint	NA		NA		NA		M	RO
docsBpi2CodeDownloadGroup								
Object	**2.0 CM in 1.0 mode**	**Access**	**2.0 CM in 1.1 mode**	**Access**	**2.0 CM in 2.0 mode**	**Access**	**CMTS**	**Access**
docsBpi2CodeDownloadStatusCode	M	RO	M		M	RO	O	RO
docsBpi2CodeDownloadStatusString	M	RO	M		M	RO	O	RO
docsBpi2CodeMfgOrgName	M	RO	M		M	RO	O	RO
docsBpi2CodeMfgCodeAccessStart	M	RO	M		M	RO	O	RO
docsBpi2CodeMfgCvcAccessStart	M	RO	M		M	RO	O	RO
docsBpi2CodeCoSignerOrgName	M	RO	M		M	RO	O	RO
docsBpi2CodeCoSignerCodeAccessStart	M	RO	M		M	RO	O	RO
docsBpi2CodeCoSignerCvcAccessStart	M	RO	M		M	RO	O	RO
docsBpi2CodeCvcUpdate	M	RW	M		M	RW	O	RW
DOCS-LOADBALANCING-MIB								
Object					**CM**	**Access**	**CMTS**	**Access**
DocsLoadBalEnable					NA	NA	M	RW
docsLoadBalChgOverGroup								
docsLoadBalChgOverMacAddress					NA	NA	M	RW
docsLoadBalChgOverDownFrequency					NA	NA	M	RW
docsLoadBalChgOverUpChannelId					NA	NA	M	RW
docsLoadBalChgOverInitTech					NA	NA	M	RW
DocsLoadBalChgOverCmd					NA	NA	M	RW
docsLoadBalChgOverCommit					NA	NA	M	RW
docsLoadBalChgOverLastCommit					NA	NA	M	RW
docsLoadBalChgOverStatusTable								
docsLoadBalChgOverStatusMacAddr					NA	NA	M	RO
docsLoadBalChgOverStatusDownFreq					NA	NA	M	RO
docsLoadBalChgOverStatusUpChnId					NA	NA	M	RO
docsLoadBalChgOverStatusInitTech					NA	NA	M	RO
docsLoadBalChgOverStatusCmd					NA	NA	M	RO
docsLoadBalChgOverStatusValue					NA	NA	M	RO
docsLoadBalChgOverStatusUpdate					NA	NA	M	RO
DocsLoadBalGrpTable								
docsLoadBalGrpId					NA	NA	M	N-Acc
docsLoadBalGrpIsRestricted					NA	NA	M	RC

docsLoadBalGrpInitTech	NA	NA	M	RC
docsLoadBalGrpDefaultPolicy	NA	NA	M	RC
docsLoadBalGrpEnable	NA	NA	M	RC
docsLoadBalGrpSuccess	NA	NA	M	RC
docsLoadBalGrpDCCFails	NA	NA	M	RC
docsLoadBalGrpStatus	NA	NA	M	RC
DocsLoadBalChannelTable				
docsLoadBalChannelIfIndex	NA	NA	M	RC
docsLoadBalChannelStatus	NA	NA	M	RC
DocsLoadBalChnPairsTable				
docsLoadBalChnPairsIfIndexDepart	NA	NA	M	N-Acc
docsLoadBalChnPairsIfIndexArrive	NA	NA	M	N-Acc
docsLoadBalChnPairsOperStatus	NA	NA	M	RO
docsLoadBalChnPairsInitTech	NA	NA	M	RC
docsLoadBalChnPairsRowStatus	NA	NA	M	RC
docsLoadBalRestrictCmTable				
docsLoadBalRestrictCmIndex	NA	NA	M	N-Acc
docsLoadBalRestrictCmMACAddr	NA	NA	M	RC
docsLoadBalRestrictCmMacAddrMask	NA	NA	M	RC
docsLoadBalRestrictCmStatus	NA	NA	M	RC
docsLoadBalPolicyTable				
docsLoadBalPolicyId	NA	NA	M	N-Acc
docsLoadBalPolicyRuleId	NA	NA	M	RC
docsLoadBalPolicyRulePtr	NA	NA	M	RC
docsLoadBalPolicyRowStatus	NA	NA	M	RC
docsLoadBalBasicRuleTable				
docsLoadBalBasicRuleId	NA	NA	M	N-Acc
docsLoadBalBasicRuleEnable	NA	NA	M	RC
docsLoadBalBasicRuleDisStart	NA	NA	M	RC
docsLoadBalBasicRuleDisEnd	NA	NA	M	RC
docsLoadBalBasicRuleRowStatus	NA	NA	M	RC

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.)

vacmContextTable

Object	CM	Access	CMTS	Access
vacmContextName	M	RO	M	RO

vacmSecurityToGroupTable

Object	CM	Access	CMTS	Access
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

vacmAccessTable

Object	CM	Access	CMTS	Access
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

vacmViewTreeFamilyTable

Object	CM	Access	CMTS	Access
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: The SNMP-COMMUNITY-MIB is ONLY accessible when the device is in SNMP Coexistence mode.

snmpCommunityTable

Object	CM	Access	CMTS	Access
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

SnmpTargetExtTable

Object	CM	Access	CMTS	Access
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.

snmpEngine Group

Object	CM	Access	CMTS	Access
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: The SNMP Message Processing and Dispatching MIB is ONLY accessible when the device is in SNMP Coexistence mode.

snmpMPDStats

Object	CM	Access	CMTS	Access
snmpUnknownSecurityModels	M	RO	M	RO
snmpInvalidMsgs	M	RO	M	RO
snmpUnknownPDUHandlers	M	RO	M	RO

SNMP Applications [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: The [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								
Object	**2.0 CMin 1.0 Mode**	**Access**	**2.0 CMin 1.1 Mode**	**Access**	**2.0 CMin 2.0 Mode**	**Access**	**CMTS**	**Access**
docsIfDocsisCapability	D	RO	D	RO	N-Sup		N-Sup	
docsIfDocsisOperMode	D	RO	D	RO	N-Sup		N-Sup	
docsIfCmtsCmStatusDocsisMode	N/A	N/A	N/A	N/A	N/A	N/A	N-Sup	
DOCS-CABLE-DEVICE-TRAP-MIB								
Object	**2.0CM in 1.0 Mode**	**Access**	**2.0 CMin 1.1 Mode**	**Access**	**2.0 CMin 2.0 Mode**	**Access**	**CMTS**	**Access**
docsDevCmTrapControl	O	RW	M	RW	M	RW	NA	
docsDevCmtsTrapControl	NA		NA		NA		M	RW
docsDevCmInitTLVUnknownTrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmDynServReqFailTrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmDynServRspFailTrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmDynServAckFailTrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmBpiInitTrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmBPKMTrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmDynamicSATrap	NA		M	ATRAP	M	ATRAP	NA	
docsDevCmDHCPFailTrap	O	ATRAP	M	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeInitTrap	O	ATRAP	M	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeFailTrap	O	ATRAP	M	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeSuccessTrap	O	ATRAP	M	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeCVCFailTrap	O	ATRAP	M	ATRAP	M	ATRAP	NA	

docsDevCmTODFailTrap	O	ATRAP	M	ATRAP	M	ATRAP	NA	
docsDevCmDCCReqFailTrap	O	ATRAP	M	ATRAP	M	ATRAP		
docsDevCmDCCRspFailTrap	O	ATRAP	M	ATRAP	M	ATRAP		
docsDevCmDCCAckFailTrap	O	ATRAP	M	ATRAP	M	ATRAP		
docsDevCmtsInitRegReqFailTrap			NA		NA		M	ATRAP
docsDevCmtsInitRegRspFailTrap			NA		NA		M	ATRAP
docsDevCmtsInitRegAckFailTrap			NA		NA		M	ATRAP
docsDevCmtsDynServReqFailTrap			NA		NA		M	ATRAP
docsDevCmtsDynServRspFailTrap			NA		NA		M	ATRAP
docsDevCmtsDynServAckFailTrap			NA		NA		M	ATRAP
docsDevCmtsBpiInitTrap			NA		NA		M	ATRAP
docsDevCmtsBPKMTrap			NA		NA		M	ATRAP
docsDevCmtsDynamicSATrap			NA		NA		M	ATRAP
docsDevCmtsDCCReqFailTrap			NA		NA		M	ATRAP
docsDevCmtsDCCRspFailTrap			NA		NA		M	ATRAP
docsDevCmtsDCCAckFailTrap			NA		NA		M	ATRAP

A.1 IF-MIB ifTable MIB-Object details

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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)	(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	205	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 by the number of bits per symbol. For RF Upstream; This is the raw band-width 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-QAM=42,884,296	(n)	(n)	10,000,000	0	~64-QAM=30,341,646 ~256-QAM=42,884,296	(n)	12,000,000	speed

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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-QAM=42	(n)*	(n)**	10	0	~64-QAM=30, ~256-QAM=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	Empty-String	Enet-MAC	CATV-MAC	Empty-String	Empty-String	USB-PhysAddr.	PhysAddr.

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
ifAdminStatus: "The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. [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 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)	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)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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	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 ifCounter DiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(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)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPEOther 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 ifCounter DiscontinuityTime."	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 ***	(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 ifCounter DiscontinuityTime."	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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 ifCounter DiscontinuityTime."	0 or (n) = 64-bit count ***	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	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 sublayer. [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 ifCounter DiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPEOther 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 ***	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 sublayer. [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)	(n)	MUST be 0	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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 ifCounter DiscontinuityTime."	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	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 ifCounter DiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPEOther 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 ifCounterDiscontinuity Time."	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 ***	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)	(n)	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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 ifCounter DiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
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 ifCounter DiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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 ifCounter DiscontinuityTime."	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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	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 ***

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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 reinitialization of the management system, and at other times as indicated by the value of ifCounter DiscontinuityTime."	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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 ifCounter DiscontinuityTime."	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 ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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 ifCounter DiscontinuityTime."	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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	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 ifCounter DiscontinuityTime."	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

IF-MIB Object details for Cable Device using 10 Mbps Ethernet	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream Logical Channel**	**CM-Ethernet-10**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)
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)	true(1) false(2)	false(2)	true(1) false(2)	true(1) false(2)

RFC-2863 MIB-Object details for Cable Device, using 100 Mbps Ethernet **(Effected MIB-Objects only; All others are the same as preceeding table)**	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream**	**CM-Ethernet-100**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther 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 reportableby 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)	(n)	100,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	12,000,000	speed
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)	(n)	100	0	~64-QAM=30, ~256-QUAM=42	(n)	12	speed

RFC-2863 MIB-Object details for Cable Device, using 100 Mbps Ethernet **(Effected MIB-Objects only; All others are the same as preceeding table)**	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream**	**CM-Ethernet-100**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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 reinitialization of the management system, and a other times, as indicated by the value of ifCounterDiscontinuityTime."	(n)= low 32-bits of the 64-bit count	(n)	MUST be 0	(n)	(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)
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 reinitialization of the management system, and a 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	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 ***

RFC-2863 MIB-Object details for Cable Device, using 100 Mbps Ethernet **(Effected MIB-Objects only; All others are the same as preceeding table)**	**CMTS-Ethernet-10**	**CMTS-MAC**	**CMTS-Downstream**	**CMTS-Upstream Physical Interface**	**CMTS-Upstream**	**CM-Ethernet-100**	**CM-MAC**	**CM-Downstream**	**CM-Upstream**	**CM-USB**	**CM-CPEOther Type**
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 data octets received from upper protocol layers and transmitted on this interface. For MAC; The total number of data octets (bridge data, data target for the managed device) transmitted on this interface from RF upstream interface and after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at reinitialization of the management system, and a 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	MUST be 0	(n)= low 32-bits of the 64-bit count	(n)	MUST be 0	(n)	(n)	(n)
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 data 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 reinitialization of the management system, and a other times, as indicated by the value of ifCounterDiscontinuityTime."	(n)= 64-bit count	(n)= 64-bit count	(n)= 64-bit count	MUST be 0	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 ***

A.2 [RFC 1493] and [RFC 2863] MIB-Object Details for CCCM

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

A.2.1 [RFC 1493] MIB-Object Details

Table A-1 [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	
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	

Table A-1 [RFC 1493] MIB-Object Details (Continued)

BRIDGE-MIB [RFC 1493]		
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

A.2.2 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 A-2 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 following Table	

Dot1dBasePortTable
The following table contains generic information about every port that is associated.

Table A-3 Dot1dBasePortTable

Mib Object	Object Valve Description	Access
Dot1dBasePort	1 - for CPE port; 2 - forRF 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 implement 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 A-4 Dot1dBasePortTable

Mib Object	Object Valve Description	Access
dot1dTpLearnedEntryDiscards	Supported	Hard-coded, read-only
dot1dTpAgingTime	1000001	Hard-coded, read-only
dot1dTpEntry (Table)	For transparent bridging only - read-only Table has 2 entries, see below	
dot1dTpFdbAddress	CPE MAC address	Hard-coded, read-only
dot1dTpFdbPort	0 (port number has not been learned)	Hard-coded, read-only
dot1dTpFdbStatus	4: self(4)	Hard-coded, read-only
dot1dTpPortTable	See Table below	

Table A-5 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

The dot1dTpPortTable

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

Table A-6 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.

A.2.3 [RFC 2863] ifTable MIB-Object details for CCCM

From SNMP perspective, CCCM MUST mimic the standalone CM. The generic network interface MIB on logical CPE interface MUST be supported [RFC 2863], with the following recommended values:

Table A-7 [RFC 2863] ifTable MIB-Object details for CCCM

ifTable / ifXTable Table Field	Implementation for CPE Interface
ifIndex	1
ifDescr	“Textual description”
ifType	1 - other
ifMtu	1500
ifSpeed	10 Mbit/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 Annex B.1
ifLastChance	Up to[RFC 2863] and OSSI Annex B.1
ifInOctets	Up to[RFC 2863] and OSSI Annex B.1
ifInUcastPkts	Up to[RFC 2863] and OSSI Annex B.1
ifInDiscards	Up to [RFC 2863] and OSSI Annex B.1
ifInErrors	Up to [RFC 2863] and OSSI Annex B.1
ifInUnknownProtos	Up to [RFC 2863] and OSSI Annex B.1
ifOutOctets	Up to [RFC 2863] and OSSI Annex B.1

Table A-7 [RFC 2863] ifTable MIB-Object details for CCCM (Continued)

ifTable / ifXTable Table Field	Implementation for CPE Interface
ifOutUcastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifOutDiscards	Up to [RFC 2863] and OSSI Annex B.1
ifOutErrors	Up to [RFC 2863] and OSSI Annex B.1
ifName	Up to[RFC 2863] and OSSI Annex B.1
ifInMulticastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifInBroadcastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifOutMulticastPkts	Up to[RFC 2863] and OSSI Annex B.1
ifOutBroadcastPkts	Up to[RFC 2863] and OSSI Annex B.1
ifHCInOctets	Up to[RFC 2863] and OSSI Annex B.1
ifHCInUcastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifHCInMulticastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifHCInBroadcastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifHCOutOctets	Up to [RFC 2863] and OSSI Annex B.1
ifHCOutUcastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifHCOutMulticastPkts	Up to [RFC 2863] and OSSI Annex B.1
ifHCOutBroadcastPkts	Up to[RFC 2863] and OSSI Annex B.1
ifLinkUpDownTrapEnable	Up to[RFC 2863] and OSSI Annex B.1, enable by default
ifHighSpeed	10 Mbit/sec
ifPromiscuousMode	TRUE, read-only access
ifConnectorPresent	Always True(1)
ifAlias	Up to [RFC 2863] and OSSI Annex B.1
ifCounterDiscontinuityTime	Up to [RFC 2863] and OSSI Appendix B

Annex B IPDR Standards Submission for DOCSIS Cable Data Systems Subscriber Usage Billing Records[1]

Note: 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 7.1 of this specification.

B.1 Service Definition

Cable Data Systems consist of Cable Modem Termination Systems (CMTSes), 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.

B.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.

1. Revised Annex B per ECN OSS2-N-02236 by GO on 02/12/03.

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.

B.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 B.2.1 and the summary table (Table 1) below.

B.1.2.1 IPDR Information

B.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.

B.1.2.1.2 seqNum

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

B.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.

B.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 />.

B.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.

B.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.

B.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.

B.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.

B.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.

B.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.

B.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.

B.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.

B.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.

B.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.

B.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.

B.1.2.4.4 SFdirection

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

B.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.

B.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.

B.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.

B.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 B-1 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

Table B-1 Service Usage Element Names (Continued)

Category	Name	Type	Presence	Possible Values	Remarks
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 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	Optional	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.)

B.2 Example IPDR XML Subscriber Usage Billing Records

The example Subscriber Usage Billing File can be viewed easily via a standard web browser (such as Microsoft Internet Explorer 6.0) if the DOCSIS schema file (DOCSIS-3.1-B.0.xsd) is placed in the same directory as the billing file.

B.2.1 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>
```

B.2.2 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>
```

This page intentionally left blank.

Annex C SNMPv2c INFORM Request Definition for Subscriber Account Management (SAM) (normative)

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

This page intentionally left blank.

Annex D Format and Content for Event, SYSLOG, and SNMP Trap (normative)

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

Each row specifies a possible event that may appear in the CM or CMTS. These events are to be reported by a cable device through local event logging, and may be accompanied by syslog or 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 the CM or CMTS. These priorities are the same is reported in the docsDevEvLevel object in the 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 cable device MIB and the text field of the syslog. The sixth column provides additional information about the event text in the fifth column. Some of the text fields 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.[1]

The next column specifies the error code. The eighth column indicates a 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 Section 7.4.2.2.2. Please note that the algorithm in Section 7.4.2.2.2 will generate a hexadecimal number. The event IDs in this list have been converted to decimal integers from the hexadecimal number.

The syslog format is specified in Section 7.4.2.2.2 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 of the form “Service Add rejected - Unspecified reason - MAC addr: x1 x2 x3 x4 x5 x6”.

The last column, “TRAP NAME”, is docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap. This indicates that the event is notified by the SNMP trap docsDevCmDynServReqFailTrap in a cable modem and docsDevCmtsDynServReqFailTrap in a CMTS.[2] [3] [4]

---

1. Added the last sentence per ECN OSS2-N-03083 by GO on 11/17/03.
2. Revised table on following page per ECN OSS2-N-02235 by GO on 02/11/03.
3. Revised Error Code Set E206.0-E208.0 per ECN OSS2-N-03010 by GO on 02/26/03.
4. Revised Process BPKM CM PRIORITY per ECN OSS2-N-03017 by GO on 03/20/03.

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
DOWNSTREAM ACQUISITION FAILED								
Init	DOWN-STREAM ACQUISI-TION	Critical		SYNC Timing Synchronization failure - Failed to acquire QAM/ QPSK symbol timing		T01.0	84000100	
Init	DOWN-STREAM ACQUISI-TION	Critical		SYNC Timing Synchronization failure - Failed to acquire FEC framing		T02.0	84000200	
Init	DOWN-STREAM ACQUISI-TION	Critical		SYNC Timing Synchronization failure, Acquired FEC framing - Failed to ac-quire MPEG2 Sync		T02.1	84000201	
Init	DOWN-STREAM ACQUISI-TION	Critical		SYNC Timing Synchronization failure - Failed to acquire MAC framing		T03.0	84000300	
Init	DOWN-STREAM ACQUISI-TION	Critical		SYNC Timing Synchronization failure - Failed to receive MAC SYNC frame within time-out period		T04.0	84000400	
Init	DOWN-STREAM ACQUISI-TION	Critical		SYNC Timing Synchronization failure - Loss of Sync		T05.0	84000500	
FAILED TO OBTAIN UPSTREAM PARAMETERS								
Init	OBTAIN UP-STREAM PARAME-TERS	Critical		No UCDs Re-ceived - Timeout		U01.0	85000100	
Init	OBTAIN UP-STREAM PARAME-TERS	Critical		UCD invalid or channel unus-able		U02.0	85000200	
Init	OBTAIN UP-STREAM PARAME-TERS	Critical		UCD & SYNC valid - NO MAPS for this channel		U04.0	85000400	

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	OBTAIN UP-STREAM PARAME-TERS	Critical		US channel wide parameters not set before Burst Descriptors		U06.0	85000600	
MAP Upstream Bandwidth Allocation								
Any	Any	informa-tional	Infor-mation-al	A transmit op-portunity was missed because the MAP arrived too late.		M01.0	77000100	
RANGING FAILED : RNG-REQ RANGING REQUEST								
Init	RANG-ING	Critical		No Maintenance Broadcasts for Ranging oppor-tunities received - T2 time-out		R01.0	82000100	
Init	RANG-ING	Critical		Received Re-sponse to Broadcast Main-tenance Re-quest, But no Unicast Mainte-nance opportuni-ties received - T4 timeout		R04.0	82000400	
Init	RANG-ING		Warn-ing	No Ranging Re-quests received from POLLED CM (CMTS gen-erated polls).		R101.0	82010100	
Init	RANG-ING		Warn-ing	Retries exhaust-ed for polled CM (report MAC ad-dress). After 16 R101.0 errors.		R102.0	82010200	
Init	RANG-ING		Warn-ing	Unable to Suc-cessfully Range CM (report MAC address) Re-tries 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	RANG-ING		Warn-ing	Failed to re-ceive Periodic RNG-REQ from modem (SID X), timing-out SID.		R104.0	82010400	
RANGING FAILED : RNG-RSP RANGING RESPONSE								
Init	RANG-ING	Critical		No Ranging Re-sponse received - T3 time-out		R02.0	82000200	

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	RANGING	Critical		Ranging Request Retries exhausted		R03.0	82000300	
Init	RANGING	Critical		Started Unicast Maintenance Ranging - No Response received - T3 timeout		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	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	Critical		DHCP FAILED - Requested Info not supported.		D03.0	68000300	
Init	DHCP	Critical		DHCP FAILED - Response doesn't contain ALL the valid fields as described in the RFI spec Annex D		D03.1	68000301	
Init	TFTP	Critical		TFTP failed - Request sent - No Response		D05.0	68000500	

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	
Init	TFTP	Critical		TFTP Failed - file too big		D10.0	68001000	
Init	TFTP	Critical		TFTP file complete- but doesn't enable 2.0 Mode - conflicts with current US channel type	For SYSLOG only: append: File name = <P1> P1 = filename of TFTP file	D11.0	68001100	
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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 TLVs	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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad US CH - Overload	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I107.0	73010700	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Network Access has Invalid Pa-rameter	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I108.0	73010800	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Class of Service - Invalid Configuration	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I109.0	73010900	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Class of Service - Unsup-ported class	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I110.0	73011000	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Class of Service - Invalid class ID or out of range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I111.0	73011100	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Max DS Bit Rate - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I112.0	73011200	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Max DS Bit Rate Unsupport-ed Setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I112.1	73011201	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Max US Bit - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I113.0	73011300	docsDevC-mtsInitRe-gReqFailTrap
Init	REGIS-TRATION RE-QUEST		Warn-ing	Bad Max US Bit Rate - Unsup-ported Setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC ad-dress	I113.1	73011301	docsDevC-mtsInitRe-gReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Version 1.1 and 2.0 Specific REG-REQ REGISTRATION REQUEST								
Init	1.1 and 2.0 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
Init	1.1 and 2.0 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 and 2.0 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 and 2.0 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 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Required parameter not present <P1>	P1 = TLV typeIt 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 and 2.0 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 and 2.0 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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	1.1 and 2.0 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 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - parameter invalid for context <P1>	P1 = TLV parameterFor CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.8	73020108	docsDevCmtsInitRegReqFailTrap
Init	1.1 and 2.0 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 and 2.0 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 and 2.0 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 and 2.0 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 and 2.0 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 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Message syntax error <P1>	P1 = messageFor CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.14	73020114	docsDevCmtsInitRegReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Primary service flow error <P1>	P1 = Service Flow ReferenceFor CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.15	73020115	docsDevCmtsInitRegReqFailTrap
	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Message too big <P1>	P1 = # of charactersFor CMTS SYSLOG only, append: MAC Addr: <P2>. P2 = CM MAC address	I201.16	73020116	docsDevCmtsInitRegReqFailTrap
REG-RSP REGISTRATION RESPONSE								
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 and 2.0 Specific REG-RSP								
Init	1.1 and 2.0 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 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains classifier parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.1	73025101	
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains PHS parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.2	73025102	

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected unspecified reason		I251.3	73025103	
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message syntax error <P1>	P1 = message	I251.4	73025104	
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message too big <P1>	P1 = # of characters	I251.5	73025105	
Version 2.0 Specific REG-RSP								
Init	2.0 SPECIFIC REGISTRATION RESPONSE	Warning		REG-RSP received after REG-ACK. Returning to 1.x transmit mode		I261.0	73026100	
REG-ACK REGISTRATION ACKNOWLEDGEMENT								
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								
Init	TLV-11 PARSING	Notice		TLV-11 - unrecognized OID		I401.0	73040100	docsDevCmInitTLVUnknownTrap
Init	TLV-11 PARSING	Critical		TLV-11 - Illegal Set operation failed		I402.0	73040200	docsDevCmInitTLVUnknownTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
Init	TLV-11 PARSING	Critical		TLV-11 - Failed to set duplicate elements		I403.0	73040300	docsDevC-mInitTLVUn-knownTrap
SW UPGRADE INIT								
SW Up-grade	SW UP-GRADE INIT	Notice		SW Download INIT - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp serv-er IP address	E101.0	69010100	docsDevCm-SwUpgra-deInitTrap
SW Up-grade	SW UP-GRADE INIT	Notice		SW Download INIT - Via Con-fig file <P1>	P1 = CM config file nameFor SYSLOG only, append: SW file: <P2> - SW server: < P3>. P2 = SW file name and P3 = Tftp server IP address	E102.0	69010200	docsDevCm-SwUpgra-deInitTrap
SW UPGRADE GENERAL FAILURE								
SW Up-grade	SW UP-GRADE GENER-AL FAIL-URE	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 serv-er IP address	E103.0	69010300	docsDevCm-SwUpgrade-FailTrap
SW Up-grade	SW UP-GRADE GENER-AL FAIL-URE	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 serv-er IP address	E104.0	69010400	docsDevCm-SwUpgrade-FailTrap
SW Up-grade	SW UP-GRADE GENER-AL FAIL-URE	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 serv-er IP address	E105.0	69010500	docsDevCm-SwUpgrade-FailTrap
SW Up-grade	SW UP-GRADE GENER-AL FAIL-URE	Error		SW upgrade Failed before download -TFTP Max Retry Ex-ceeded	For SYSLOG only, append: SW file: <P1> - SW server: < P2>. P1 = SW file name and P2 = Tftp serv-er IP address	E106.0	69010600	docsDevCm-SwUpgrade-FailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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
DHCP FAILURE AFTER CM HAS REGISTERED WITH THE CMTS								
DHCP		Error		DHCP RENEW sent - No response		D101.0	68010100	docsDevCmDHCPFailTrap
DHCP		Error		DHCP REBIND sent - No response		D102.0	68010200	docsDevCmDHCPFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
DHCP		Error		DHCP RENEW sent - Invalid DHCP option		D103.0	68010300	docsDevCm-DHCPFail-Trap
DHCP		Error		DHCP REBIND sent - Invalid DHCP option		D104.0	68010400	docsDevCm-DHCPFail-Trap
DYNAMIC SERVICE REQUEST								
DY-NAMIC SER-VICES	DYNAM-IC SER-VICE RE-QUEST	Error	Warn-ing	Service Add re-jected - Unspeci-fied reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.0	83000100	docsDevCm-DynServReq-FailTrap, docsDevC-mtsDyn-ServReqFail-Trap
DY-NAMIC SER-VICES	DYNAM-IC SER-VICE RE-QUEST	Error	Warn-ing	Service Add re-jected - Unrec-ognized configu-ration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.1	83000101	docsDevCm-DynServReq-FailTrap, docsDevC-mtsDyn-ServReqFail-Trap
DY-NAMIC SER-VICES	DYNAM-IC SER-VICE RE-QUEST	Error	Warn-ing	Service Add re-jected - Tempo-rary no resource	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.2	83000102	docsDevCm-DynServReq-FailTrap, docsDevC-mtsDyn-ServReqFail-Trap
DY-NAMIC SER-VICES	DYNAM-IC SER-VICE RE-QUEST	Error	Warn-ing	Service Add re-jected - Perma-nent administra-tive	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.3	83000103	docsDevCm-DynServReq-FailTrap, docsDevC-mtsDyn-ServReqFail-Trap
DY-NAMIC SER-VICES	DYNAM-IC SER-VICE RE-QUEST	Error	Warn-ing	Service Add re-jected - Re-quired parame-ter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.4	83000104	docsDevCm-DynServReq-FailTrap, docsDevC-mtsDyn-ServReqFail-Trap
DY-NAMIC SER-VICES	DYNAM-IC SER-VICE RE-QUEST	Error	Warn-ing	Service Add re-jected - Header suppression set-ting not support-ed	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.5	83000105	docsDevCm-DynServReq-FailTrap, docsDevC-mtsDyn-ServReqFail-Trap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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 Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, 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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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 - MAC-addr: <P1	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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected -Duplicate reference_ID or index inmessage	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.11	83010111	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected -Classifier for another flow - MACaddr: <P1>	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.12	83010112	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected -Message too big	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.18	83010118	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
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	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICE ACKNOWLEDGEMENTS								
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	DocsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap
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	docsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap
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	docsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap
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	docsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap
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	docsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap
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	docsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap
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	docsDevC-mBPK-MTrap, docs-DevCmtsBPKMTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
EVENT BETWEEN AUTH & TEK FSM								
BPKM		Informational		Authorized	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B401.0	66040100	docsDevCmBPKMTrap
BPKM		Informational		Auto Pend	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B402.0	66040200	docsDevCmBPKMTrap
BPKM		Informational		Auth Comp	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B403.0	66040300	docsDevCmBPKMTrap
BPKM		Informational		Stop	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B404.0	66040400	docsDevCmBPKMTrap
TEK FSM								
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
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	docsDevCmDynamicSATrap
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	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
Dynamic SA		Error		Map Request Retry Timeout	For CM SYSLOG only append: MAC addr: <P1>. P1 = Mac Addr of CMTS	B603.0	66060300	docsDevCmDynamicSATrap
Dynamic SA		Informational		Unmap	For CM SYSLOG only append: MAC addr: <P1>. P1 = Mac Addr of CMTS	B604.0	66060400	docsDevCmDynamicSATrap
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	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
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	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
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	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
VERIFICAITON 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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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	
UCC-RSP Upstream Channel Change Response								

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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
DCC	DCC Request	Error	Warning	DCC rejected message too big		C218.0	67021800	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Error	Warning	DCC rejected 2.0 mode disabled		C219.0	67021900	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

Process	Sub-Process	CM Priority	CMTS Priority	Event Messsage	Message Notes And Details	Error Code Set	Event ID	Trap Name
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								
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

Annex E Application of [RFC 2933] to DOCSIS 2.0 Active/Passive IGMP Devices (normative)

E.1 DOCSIS 2.0 IGMP MIBs

DOCSIS 2.0 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 2.0 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 2.0 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 2.0 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 2.0 devices.

E.1.1 IGMP Capabilities: Active and Passive Mode

There are two basic modes of IGMP capability that are applicable to a DOCSIS 2.0 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 VI of [DOCSIS 5]). In passive mode, the device derives its IGMP timers based on the rules specified in section 5.3.1 of the RFI specification. 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 2.0 specification, the device MUST meet the requirements stated in section 5.3.1 of [DOCSIS 5] and MUST support the IDMR IGMP MIB as described herein. As presently specified in the DOCSIS 2.0, 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.

E.1.2 IGMP Interfaces

A description of the application of the IDMR IGMP MIB to DOCSIS 2.0 devices follows. This description is organized by CM and CMTS device type.

E.2 DOCSIS 2.0 CM Support for the IGMP MIB

There are two types of interfaces applicable to IGMP on the DOCSIS 2.0 CM. These are the HFC-Side and CMCI-Side interfaces, respectively. Application of the IGMP MIB to DOCSIS 2.0 CMs is presented in terms of passive and active CM operation and these two interface types.

E.2.1 igmpInterfaceTable- igmpInterfaceEntry

E.2.1.1 igmpInterfaceIfIndex

The ifIndex value of the interface for which IGMP is enabled.

E.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.

E.2.1.2 igmpInterfaceQueryInterval

The frequency at which IGMP Host-Query packets are transmitted on this interface.

E.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.

E.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 = (2^32-1); Default = 125

E.2.1.3 igmpInterfaceStatus

The activation of a row enables IGMP on the interface. The destruction of a row disables IGMP on the interface.

E.2.1.3.1 All Modes

MUST be enabled on both interfaces for all DOCSIS 2.0 CM interfaces.

E.2.1.4 igmpInterfaceVersion

The version of IGMP which is running on this interface. MUST be version 2 for all DOCSIS 2.0 CM interfaces.

E.2.1.5 igmpInterfaceQuerier

The address of the IGMP Querier on the IP subnet to which this interface is attached.

E.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.

E.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 CMs that participate in IGMP Querier negotiation on the CMCI may report it as a different CPE.

E.2.1.6 igmpInterfaceQueryMaxResponseTime

The maximum query response time advertised in IGMPv2 queries on this interface.

E.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

E.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.

E.2.1.7 igmpInterfaceQuerierUpTime

The time since igmpInterfaceQuerier was last changed.

E.2.1.7.1 PassiveMode

HFC-side: read-only.

CMC-side: n/a, read-only. Return a value of zero.

E.2.1.7.2 Active Mode

HFC-side: read-only.

CMCI-side: read-only.

E.2.1.8 igmpInterfaceQuerierExpiryTime

The amount of time remaining before the other querier present timer expires. If the local system is the queier, the value of this object is zero.

E.2.1.8.1 Passive Mode

Both interfaces: n/a, read-only. The CM is never the querier, return 0.

E.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.

E.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.

E.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.

E.2.1.9.2 Active Mode

HFC-side: read-only.

CMCI-side: read-only.

E.2.1.10 igmpInterfaceWrongVersionQueries

The number of queries recived 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 2.0 requires that all CM and CMTS devices support IGMPv2, it is possible for an upstream querier to be an IGMPv1 querier.

E.2.1.10.1 All Modes

All interfaces: read-only. The number of non-v2 queries received on this interface.

E.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.

E.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.

E.2.1.12.1 Passive Mode

All Interfaces: read-only. Always return a value of zero.

E.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.

E.2.1.13 igmpInterfaceGroups

The current number of entries for this interface in the Cache Table.

E.2.1.13.1 All HFC-side: n/a, read-only. Always return a value of zero (see CMCI-side).

E.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.

E.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.

E.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.

E.2.1.14.2 Active Mode

All interfaces: read-create. Min = 1; Max = (232-1); Default = 2

E.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.

E.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

E.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.

E.2.2 igmpCacheTable - igmpCacheEntry

E.2.2.1 igmpCacheAddress

The IP multicast group address for which this entry contains information.

E.2.2.1.1 All Modes

Not-accessible (index). Report the address of active IP Multicast on the CMCI interface.

E.2.2.2 igmpCacheIfIndex

The interface for which this entry contains information for an IP multicast group address.

E.2.2.2.1 All Modes

MUST only apply to CMCI interface (e.g., membership is only active on subscriber side of CM).

E.2.2.3 igmpCacheSelf

An indication of whether the local system is a member of this group address on this interface.

E.2.2.3.1 Passive Mode

Read-only. MUST be set to FALSE. The CM is not a member of any group.

E.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.

E.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.

E.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).

E.2.2.5 igmpCacheUpTime

The time elapsed since this entry was created.

E.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).

E.2.2.6 igmpCacheExpiryTime

The minimum amount of time remaining before this entry will be aged out.

E.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).

E.2.2.7 igmpCacheStatus

The status of this entry.

E.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.

E.2.2.8 igmpCacheVersion1HostTimer

The time remaining until the local querier will assume that there are no longer any IGMP version 1 members on thie 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.

E.2.2.8.1 Passive Mode

All interfaces: n/a, read-only. Return a value of zero.

E.2.2.8.2 Active Mode

HFC-side: n/a, read-only. Return a value of zero.

CMCI-side: read-only.

E.3 Docsis 2.0 CMTS support for the IGMP MIB

There are two types of interfaces applicable to IGMP on the DOCSIS 2.0 CMTS. These are the NSI-Side and NSI-Side interfaces, respectively. Application of the IGMP MIB to DOCSIS 2.0 CMTSes 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.

E.3.1 igmpInterfaceTable- igmpInterfaceEntry

E.3.1.1 igmpInterfaceIfIndex

The ifIndex value of the interface for which IGMP is enabled.

E.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.

E.3.1.2 igmpInterfaceQueryInterval

The frequency at which IGMP Host-Query packets are transmitted on this interface.

E.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.

E.3.1.2.2 Active Mode

NSI-side: read-create. Min = 0; Max = (2^32-1); Default = 125

HFC-side: read-create. Min = 0; Max = (2^32-1); Default = 125

E.3.1.3 igmpInterfaceStatus

E.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.

E.3.1.4 igmpInterfaceVersion

The version of IGMP which is running on this interface. MUST be version 2 for all DOCSIS 2.0 CMTS interfaces.

E.3.1.5 igmpInterfaceQuerier

The address of the IGMP Querier on the IP subnet to which this interface is attached.

E.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.

E.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's HFC-side interface. That is, queries MUST not originate from CMs or CPE.

E.3.1.6 igmpInterfaceQueryMaxResponseTime

The maximum query response time advertised in IGMPv2 queries on this interface.

E.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.

E.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.

E.3.1.7 igmpInterfaceQuerierUpTime

The time since igmpInterfaceQuerier was last changed.

E.3.1.7.1 PassiveMode

NSI-side: read-only.

HFC-side: n/a, read-only. Return a value of zero.

E.3.1.7.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

E.3.1.8 igmpInterfaceQuerierExpiryTime

The amount of time remaining before the other querier present timer expires. If the local system is the queier, the value of this object is zero.

E.3.1.8.1 Passive Mode

Both interfaces: n/a, read-only. The CMTS is not the querier, return 0.

E.3.1.8.2 Active Mode

NSI-side: read-only.

HFC-side: read-only. The CMTS MUST be the only querier on the HFC.

E.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.

E.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.

E.3.1.9.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

E.3.1.10 igmpInterfaceWrongVersionQueries

The number of queries recived 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 2.0 requires that all CMTS and CMTSTS devices support IGMPv2, it is possible for a network side querier to be an IGMPv1 querier.

E.3.1.10.1 All Modes

All interfaces: read-only. The number of non-v2 queries received on this interface.

E.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.

E.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.

E.3.1.11.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

E.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.

E.3.1.12.1 Passive Mode

All Interfaces: read-only. Always return a value of zero.

E.3.1.12.2 Active Mode

NSI-side: read-only.

HFC-side: read-only. Always return an ifIndex for a NSI-side interface.

E.3.1.13 igmpInterfaceGroups

The current number of entries for this interface in the Cache Table.

E.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.

E.3.1.13.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

E.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.

E.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.

E.3.1.14.2 Active Mode

All interfaces: read-create. Min = 1; Max = (2^32-1); Default = 2

E.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.

E.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.

E.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.

E.3.2 igmpCacheTable - igmpCacheEntry

E.3.2.1 igmpCacheAddress

The IP multicast group address for which this entry contains information.

E.3.2.1.1 All Modes

Not-accessible (index). Report the address of active IP Multicast on the interface.

E.3.2.2 igmpCacheIfIndex

The interface for which this entry contains information for an IP multicast group address.

E.3.2.2.1 Passive Mode

MUST only apply to HFC side interface (e.g., membership is only active on subscriber side of CMTS).

E.3.2.2.2 Active Mode

NSI-side: not-accessible

HFC-side: not-accessible

E.3.2.3 igmpCacheSelf

An indication of whether the local system is a member of this group address on this interface.

E.3.2.3.1 Passive Mode

read-only. MUST be set to FALSE. The CMTS is not a member of any group.

E.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.

E.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.

E.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).

E.3.2.4.2 Active Mode

NSI-side: read-only

HFC-side: read-only

E.3.2.5 igmpCacheUpTime

The time elapsed since this entry was created.

E.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).

E.3.2.5.2 Active Mode

NSI-side: read-only

HFC-side: read-only

E.3.2.6 igmpCacheExpiryTime

The minimum amount of time remaining before this entry will be aged out.

E.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).

E.3.2.6.2 Active Mode

NSI-side: read-only

HFC-side: read-only

E.3.2.7 igmpCacheStatus

The status of this entry.

E.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.

E.3.2.7.2 Active Mode

NSI-side: read-create

HFC-side: read-create

E.3.2.8 igmpCacheVersion1HostTimer

The time remaining until the local querier will assume that there are no longer any IGMP version 1 members on thie 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.

E.3.2.8.1 Passive Mode

All interfaces: n/a, read-only. Return a value of zero.

E.3.2.8.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

E.3.3 IGMP MIB Compliance

E.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}
```

E.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}
```

E.3.4 MIB Groups

See IGMP MIB for a description of the objects included in each group.

E.3.4.1 igmpV2HostMIBGroup

Active Devices only (optional - see notes for igmpCacheSelf).

E.3.4.2 igmpV2RouterMIBGroup

Active and Passive Devices

E.3.4.3 igmpBaseMIBGroup

Active and Passive Devices

E.3.4.4 igmpV2RouterMIBGroup

Active and Passive Devices

E.3.4.5 igmpRouterMIBGroup

Active and Passive Devices

E.3.4.6 igmpV2HostOptMIBGroup

Active and Passive Devices

E.3.4.7 igmpV2ProxyMIBGroup

Active Devices only.

Annex F Expected Behaviors for DOCSIS 2.0 Modem in 1.0, 1.1, and 2.0 Modes in OSS Area (normative)

The following table identifies DOCSIS OSSI 2.0 CM features that MAY and MUST be implemented in 1.1 or 1.0 mode.

Specific requirement	Required behavior, DOCSIS 2.0 Modem in 1.0 Mode	Required behavior, DOCSIS 2.0 Modem in 1.1 Mode	Required behavior, DOCSIS 2.0 Modem in 2.0 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.	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 2.0 specification.	CM MUST support DOCSIS standard events defined in the OSSI 2.0 specification.	CM MUST support DOCSIS standard events defined in the OSSI 2.0 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.	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 2.0 specification.	CM MUST support event level definitions specified by the OSSI 2.0 specification.	CM MUST support event level definitions specified by the OSSI 2.0 specification.

Specific requirement	Required behavior, DOCSIS 2.0 Modem in 1.0 Mode	Required behavior, DOCSIS 2.0 Modem in 1.1 Mode	Required behavior, DOCSIS 2.0 Modem in 2.0 Mode
Event storage in docsDevEventTable	Each entry in the dosDevEventTable contains an event-ID (identical to the Eventid requirement specified in Section 7.4.2.2.2), 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 than 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 7.4.2.2.2), 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 than 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 7.4.2.2.2), 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 than 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 MAY support a minimum of 10 rows of docsDevEventTable.	CM MAY support a minimum of 10 rows of docsDevEventTable.
Event log persistence	Event log MUST persist across reboots	Event log MUST persist across reboots.	Event log MUST persist across reboots.
SNMP Version of Trap Control (when CM is in SNMP v1/v2c DocsDevNm Access 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.	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> textOR<level>Cablemodem [vendor]: text	CM MUST support the syslog message with the format: <level>CABLEMODEM [vendor]: <eventId> 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 5.2 of the OSSI.	CM MUST support SNMP v1/v2c and SNMPv3 with DH	CM MUST support SNMP v1/v2c and SNMPv3 with DH
MIBs to implement	CM MUST support MIB objects as specified by Annex A.	CM MUST support MIB objects as specified by Annex A.	CM MUST support MIB objects as specified by Annex A.

Specific requirement	Required behavior, DOCSIS 2.0 Modem in 1.0 Mode	Required behavior, DOCSIS 2.0 Modem in 1.1 Mode	Required behavior, DOCSIS 2.0 Modem in 2.0 Mode
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.	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 7.2 of the OSSI 2.0 specification.	CM MUST support configuration management requirement as specified by Section 7.2 of the OSSI 2.0 specification.	CM MUST support configuration management requirement as specified by Section 7.2 of the OSSI 2.0 specification.
IP/LLC filters	CM SHOULD support LLC/IP filter requirement as specified by OSSI 2.0 specification.	CM MUST support LLC/IP filter requirement as specified by OSSI 2.0 specification.	CM MUST support LLC/IP filter requirement as specified by OSSI 2.0 specification.
CM interaction with CM configuration file	CM MUST process TLV type 11 entries in a configuration file as specified by Section 6.4 of the OSSI 2.0 specification.	CM MUST process TLV type 11 entries in a configuration file as specified by Section 6.4 of the OSSI 2.0 specification.	CM MUST process TLV type 11 entries in a configuration file as specified by Section 6.4 of the OSSI 2.0 specification.
Additional MIB objects requirement	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 6.3 of the OSSI 2.0 specification.	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 6.3 of the OSSI 2.0 specification.	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 6.3 of the OSSI 2.0 specification.
Performance management	CM MUST support performance management requirements as specified by Section 7.5 of the OSSI 2.0 specification.	CM MUST support performance management requirements as specified by Section 7.5 of the OSSI 2.0 specification.	CM MUST support performance management requirements as specified by Section 7.5 of the OSSI 2.0 specification.
OSS for CMCI	CM MUST support CMCI requirements as specified by Section 9 of the OSSI 2.0 specification.	CM MUST support CMCI requirements as specified by Section 9 of the OSSI 2.0 specification.	CM MUST support CMCI requirements as specified by Section 9 of the OSSI 2.0 specification.

This page intentionally left blank.

Annex G DOCS-IF-EXT-MIB (normative)

All objects included in the DOCS-IF-EXT-MIB have corresponding objects in the MIB specified in DOCS-RFI-MIB .

A 2.0 CMTS and a 2.0 CM in 2.0 mode MUST implement both DOCS-IF-EXT-MIB and DOCS-RFI-MIB.

It is intended that an ECN will be released later requiring all CMs in all modes to support DOCS-RFI-MIB and deprecate DOCS-IF-EXT-MIB.

This MIB extends the [RFC 2670] 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
```

This page intentionally left blank.

Annex H DOCS-CABLE-DEVICE-TRAP-MIB (normative)

```
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-rfmibv2-02

docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
docsIfCmtsCmStatusDocsisMode -- deprecated
FROM DOCS-IF-EXT-MIB -- deprecated

ifPhysAddress
FROM IF-MIB;

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 connected.
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 [RFC 2669]. 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
```

Annex I Requirements for DOCS-LOADBALANCING-MIB (mandatory)[1]

```
DOCS-LOADBALANCING-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    Integer32,
    Counter32,
    zeroDotZero
        FROM SNMPv2-SMI
    TruthValue,
    MacAddress,
    RowStatus,
    RowPointer,
    TimeStamp,
    TEXTUAL-CONVENTION
        FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    InterfaceIndex
        FROM IF-MIB
    docsIfCmtsCmStatusEntry,
    docsIfCmtsCmStatusIndex
        FROM DOCS-IF-MIB
    clabProjDocsis
        FROM CLAB-DEF-MIB;

docsLoadBalanceMib MODULE-IDENTITY
    LAST-UPDATED    "200403101700Z" -- March 10, 2004
    ORGANIZATION    "Cable Television Laboratories, Inc"
    CONTACT-INFO
        "
         Postal: Cable Television Laboratories, Inc.
         400 Centennial Parkway
         Louisville, Colorado 80027-1266
         U.S.A.
         Phone: +1 303-661-9100
         Fax: +1 303-661-9199
         E-mail: docis-dcc@cablelabs.com
                 mibs@cablelabs.com"
    DESCRIPTION
        "This is the MIB Module for the load balancing.
         Load balancing is manageable on a per-CM basis.
         Each CM is assigned:
            a) to a set of channels (a Load Balancing Group) among
               which it can be moved by the CMTS
            b) a policy which governs if and when the CM can be moved
            c) a priority value which can be used by the CMTS in order
               to select CMs to move."
    REVISION "200403101700Z"
    DESCRIPTION
        "Initial version of this mib module."
::= { clabProjDocsis 2 }
-- ---------------------------------------------------------
-- Textual Conventions
-- ---------------------------------------------------------
```

1. Added new Annex per ECN OSSSIv2.0-N-04.0126-6 by GO on 3/16/04.

```
ChannelChgInitTechMap   ::=  TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "This textual convention enumerates the Initialization
         techniques for Dynamic Channel Change (DCC). The techniques
         are represented by the 5 most significant bits (MSB).
         Bits 0 through 4 map to initialization techniques 0 through 4.
         Each bit position represents the internal associated technique
         as described below:

         reinitializeMac(0)     : Reinitialize the MAC
         broadcastInitRanging(1): Perform Broadcast initial
                                  ranging on new channel before
                                  normal operation
         unicastInitRanging(2)  : Perform unicast ranging on new
                                  channel before normal operation
         initRanging(3)         : Perform either broadcast or
                                  unicast ranging on new channel before
                                  normal operation
         direct(4)              : Use the new channel(s) directly
                                  without re-initializing or ranging

         Multiple bits selection in 1's means the CMTS selects the best
         suitable technique among the selected in a propietary manner.
         An empty value or a value with all bits in '0' means no channel changes
         allowed"
    SYNTAX BITS {
        reinitializeMac(0),
        broadcastInitRanging(1),
        unicastInitRanging(2),
        initRanging(3),
        direct(4)
    }

-- ---------------------------------------------------------------------
-- Main Groups
-- ---------------------------------------------------------------------
docsLoadBalNotifications   OBJECT IDENTIFIER ::= { docsLoadBalanceMib 0}
docsLoadBalMibObjects      OBJECT IDENTIFIER ::= { docsLoadBalanceMib 1}
docsLoadBalSystem          OBJECT IDENTIFIER ::= { docsLoadBalMibObjects 1 }
docsLoadBalChgOverObjects  OBJECT IDENTIFIER ::= { docsLoadBalMibObjects 2 }
docsLoadBalGrpObjects      OBJECT IDENTIFIER ::= { docsLoadBalMibObjects 3 }
docsLoadBalPolicyObjects   OBJECT IDENTIFIER ::= { docsLoadBalMibObjects 4 }
docsLoadBalChgOverGroup    OBJECT IDENTIFIER ::= { docsLoadBalChgOverObjects 1 }

docsLoadBalEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Setting this object to true(1) enables internal
         autonomous load balancing operation on this CMTS.
         Setting it to false(2) disables the autonomous
         load balacing operations.
         However moving a cable modem via docsLoadBalChgOverTable
         is allowed even when this object is set to false(2)."
::= { docsLoadBalSystem 1 }

-- ---------------------------------------------------------------------
-- CMTS Cable Modem channel change operation table and related
-- objects.
-- This group of objects determines the DCC API for execution of DCC/UCC
-- commands The status of execution is reported in docsLoadBalChgOverStatusTable
-- A CMTS operator may perform downstream/upstream load balancing
```

```
-- or failure recovery using docsLoadBalChgOver parameters and red the status .
-- ---------------------------------------------------------------------------

docsLoadBalChgOverMacAddress OBJECT-TYPE
    SYNTAX     MacAddress
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The mac address of the cable modem that the CMTS
         instructs to move to a new downstream frequency and/or
         upstream channel."
    DEFVAL { '000000000000'h }
    ::= { docsLoadBalChgOverGroup 1 }

docsLoadBalChgOverDownFrequency  OBJECT-TYPE
    SYNTAX     Integer32 (0..1000000000)
    UNITS      "hertz"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The new downstream frequency to which the cable modem is
         instructed to move.  The value 0 indicates that the CMTS
         does not create a TLV for the  downstream frequency in the
         DCC-REQ message. This object has no meaning when executing
         UCC operations."
    DEFVAL { 0 }
    ::= { docsLoadBalChgOverGroup 2 }

docsLoadBalChgOverUpChannelId  OBJECT-TYPE
    SYNTAX     Integer32 (-1..255)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The new upstream channel ID to which the cable modem is
         instructed to move.  The value -1 indicates that the CMTS
         does not create a TLV for the upstream channel ID in the
         channel change request."
    DEFVAL { -1 }
    ::= { docsLoadBalChgOverGroup 3 }

docsLoadBalChgOverInitTech OBJECT-TYPE
    SYNTAX     ChannelChgInitTechMap
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The initialization technique that the cable modem is
         instructed to use when performing change over operation.

         By default this object is initialized with all the defined
         bits having a value of '1'."
    ::= { docsLoadBalChgOverGroup 4 }

docsLoadBalChgOverCmd OBJECT-TYPE
    SYNTAX INTEGER {
        any(1),
        dcc(2),
        ucc(3)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The change over command that the CMTS is instructed
         use when performing change over operation.

         The any(1) value indicates that the CMTS is to use its
```

```
         own algorithm to determine the appropriate command."
    DEFVAL { any }
    ::= { docsLoadBalChgOverGroup 5 }

docsLoadBalChgOverCommit  OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The command to execute the DCC/UCC operation when set to
         true(1).
         The following are reasons for rejecting an SNMP
         SET to this object:
           - The MAC address in docsLoadBalChgOverMacAddr is not an
             existing MAC address in docsIfCmtsMacToCmEntry.
           - docsLoadBalChgOverCmd is ucc(3) and
             docsLoadBalChgOverUpChannelId is '-1',
           - docsLoadBalChgOverUpChannelId is '-1' and
             docsLoadBalChgOverDownFrequency is '0'.
           - DCC/UCC operation is currently being executed for the cable modem,
             on which the new command is committed, specifically if the value of
             docsLoadBalChgOverStatusValue is one of:
             messageSent(1),
                 modemDeparting(4),
                 waitToSendMessage(6).
           - An UCC operation is commited for a non-existing upstream
             channel ID or the corresponding ifOperStatus is down(2).
           - A DCC operation is committed for an invalid or non-existing
             downstream frequency, or the corresponding ifOperStatus is
             down(2).
         In those cases, the SET is rejected with an error code
         'commitFailed'.

         After processing the SNMP SET the information in
         docsLoadBalChgOverGroup is updated in a corresponding
         entry in docsLoadBalChgOverStatusEntry.
         Reading this object always returns false(2)."
    REFERENCE
        "Data-Over-Cable Service Interface Specifications: Radio
         Frequency Interface Specification SP-RFIv2.0-I04-030730,
         Sections C.4.1, 11.4.5.1."
    DEFVAL {false}
    ::= { docsLoadBalChgOverGroup 6 }

docsLoadBalChgOverLastCommit  OBJECT-TYPE
    SYNTAX TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
         "The value of sysUpTime when docsLoadBalChgOverCommit was
         last set to true. Zero if never set."
    ::= { docsLoadBalChgOverGroup 7 }

-- ---------------------------------------------------------------------
-- CMTS Cable Modem channel change operation Status table and related
-- objects.
-- This table is an AUGMENT of docsIfCmtsCmstatusTable
-- A CMTS operator may perform downstream/upstream load balancing
-- or failure recovery using docsLoadBalChgOverTable.
-- ---------------------------------------------------------------------

docsLoadBalChgOverStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsLoadBalChgOverStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
        "A table of CMTS operation entries to reports the status
         of cable modems instructed to move to a new downstream and/or
         upstream channel. using the docsLoadBalChgOverGroup objects.

         An entry in this table is created or updated for the entry
         with docsIfCmtsCmStatusIndex that correspond to the cable modem
         MAC address of the Load Balancing operation.
         docsLoadBalChgOverCommit to true(1)."
    ::= { docsLoadBalChgOverObjects 2 }

docsLoadBalChgOverStatusEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalChgOverStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A CMTS operation entry to instruct a cable modem to move to
         a new downstream frequency and/or upstream channel.

         An operator can use this to initiate an operation
         in CMTS to instruct the selected cable modem to move to a new
         downstream frequency and/or upstream channel."
    INDEX { docsIfCmtsCmStatusIndex }
    ::= { docsLoadBalChgOverStatusTable 1 }

DocsLoadBalChgOverStatusEntry ::= SEQUENCE {
    docsLoadBalChgOverStatusMacAddr     MacAddress,
    docsLoadBalChgOverStatusDownFreq  Integer32,
    docsLoadBalChgOverStatusUpChnId    Integer32,
    docsLoadBalChgOverStatusInitTech       ChannelChgInitTechMap,
    docsLoadBalChgOverStatusCmd            INTEGER,
    docsLoadBalChgOverStatusValue          INTEGER,
    docsLoadBalChgOverStatusUpdate         TimeStamp
    }

docsLoadBalChgOverStatusMacAddr OBJECT-TYPE
    SYNTAX     MacAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The mac address set in docsLoadBalChgOverMacAddress."
    ::= { docsLoadBalChgOverStatusEntry 1 }

docsLoadBalChgOverStatusDownFreq  OBJECT-TYPE
    SYNTAX     Integer32 (0..1000000000)
    UNITS      "hertz"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The Downstream frequency set in
         docsLoadBalChgOverDownFrequency."
    DEFVAL { 0 }
    ::= { docsLoadBalChgOverStatusEntry 2 }

docsLoadBalChgOverStatusUpChnId  OBJECT-TYPE
    SYNTAX     Integer32 (-1..255)
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The upstream channel ID set in
         docsLoadBalChgOverUpChannelId."
    DEFVAL { -1 }
    ::= { docsLoadBalChgOverStatusEntry 3 }

docsLoadBalChgOverStatusInitTech OBJECT-TYPE
```

```
    SYNTAX     ChannelChgInitTechMap
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The initialization technique set in
         docsLoadBalChgOverInitTech."
    ::= { docsLoadBalChgOverStatusEntry 4 }

docsLoadBalChgOverStatusCmd OBJECT-TYPE
    SYNTAX INTEGER {
        any(1),
        dcc(2),
        ucc(3)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The load balancing command set in
        docsLoadBalChgOverCmd."
    DEFVAL { any }
    ::= { docsLoadBalChgOverStatusEntry 5 }

-- May use a textual convention to report this value
-- since is repeated for two objects

docsLoadBalChgOverStatusValue  OBJECT-TYPE
    SYNTAX INTEGER {
        messageSent(1),
        noOpNeeded(2),
        modemDeparting(3),
        waitToSendMessage(4),
        cmOperationRejected(5),
        cmtsOperationRejected(6),
        timeOutT13(7),
        timeOutT15(8),
        rejectinit(9),
        success(10)
        }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The status of the specified DCC/UCC operation.
         The enumerations are:
             messageSent(1):
                 The CMTS has sent change over request message to the
                 cable modem.
             noOpNeed(2):
                 A operation was requested in which neither the
                 DS Frequency nor the Upstream Channel ID was changed.
                 An active value in this entry's row status indicates
                 that no CMTS operation is required.
             modemDeparting(3):
                 The cable modem has responded with a change over response
                 of either a DCC-RSP with a confirmation code of depart(180)
                 or a UCC-RSP.
             waitToSendMessage(4):
                 The specified operation is active and CMTS is waiting
                 to send the channel change message with channel info to
                 the cable modem.
             cmOperationRejected(5):
                 Channel Change (such as DCC or UCC) operation was rejected
                 by the cable modem.
             cmtsOperationRejected(6)
                 Channel Change (such as DCC or UCC) operation was rejected
                 by the Cable modem Termination System.
```

```
            timeOutT13(7):
                Failure due to no DCC-RSP with confirmation code
                depart(180) received prior to expiration of the
                T13 timer.
            timeOutT15(8):
                T15 timer timed out prior to the arrival of a
                bandwidth request, RNG-REQ message, or DCC-RSP message
                with confirmation code of arrive(181) from the
                cable modem.
            rejectInit(9):
                DCC operation rejected due to unsupported
                initialization tech requested.
            success(10):
                CMTS received an indication that the CM successfully
                completed the change over operation.
                E.g. If an initialization technique of re-initialize the
                MAC is used, success in indicated by the receipt
                of a DCC-RSP message with a confirmation code of
                depart(180).  In all other cases, success is
                indicated by:
                    (1) the CMTS received a DCC-RSP message with
                        confirmation code of arrive(181)
                or
                    (2) the CMTS internally confirms the presence
                        of the CM on the new channel."
    REFERENCE
        "Data-Over-Cable Service Interface Specifications: Radio
         Frequency Interface Specification SP-RFIv2.0-I04-030730,
         Sections C.4.1, 11.4.5.1."
    DEFVAL { waitToSendMessage }
    ::= { docsLoadBalChgOverStatusEntry 6 }

docsLoadBalChgOverStatusUpdate OBJECT-TYPE
    SYNTAX TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The value of sysUpTime when docsLoadBalChgOverStatusValue
         was last updated."
    ::= { docsLoadBalChgOverStatusEntry 7 }

-- ---------------------------------------------------------------------
-- Load balancing group is a cluster of downstream and associated
-- upstream channels, among which modems that are registered
-- on any of those channels, can be load balanced.
--
-- There are two types of Load Balancing Groups, General Load Balancing
-- Groups and Restricted Load Balancing Groups.  A Restricted Load
-- Balancing Group is associated with a specific, provisioned set of
-- cable modems while General Load Balancing Groups are open for CMs
-- which are not provisioned into a Restricted Load Balancing Group.
-- ---------------------------------------------------------------------

docsLoadBalGrpTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF DocsLoadBalGrpEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table contains the attributes of the load balancing
         groups present in this CMTS."
    ::= { docsLoadBalGrpObjects 1 }

docsLoadBalGrpEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalGrpEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
        "A set of attributes of load balancing group in the CMTS.
         It is index by a docsLoadBalGrpId which is unique
         within a CMTS.
         Entries in this table persist after CMTS initialization."
    INDEX { docsLoadBalGrpId }
    ::= { docsLoadBalGrpTable 1 }

DocsLoadBalGrpEntry ::= SEQUENCE {
    docsLoadBalGrpId               Unsigned32,
    docsLoadBalGrpIsRestricted     TruthValue,
    docsLoadBalGrpInitTech         ChannelChgInitTechMap,
    docsLoadBalGrpDefaultPolicy    Unsigned32,
    docsLoadBalGrpEnable           TruthValue,
    docsLoadBalGrpChgOverSuccess   Counter32,
    docsLoadBalGrpChgOverFails     Counter32,
    docsLoadBalGrpStatus           RowStatus
    }

docsLoadBalGrpId OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique index assigned to the load balancing
         group by the CMTS."
    ::= { docsLoadBalGrpEntry 1 }

docsLoadBalGrpIsRestricted OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A value true(1)Indicates type of load balancing group.
         A Restricted Load Balancing Group is associated to a specific
         provisioned set of cable modems.  Restricted Load Balancing
         Group is used to accommodate a topology specific or provisioning
         specific restriction.  Example such as a group that are reserved
         for business customers).

         Setting this object to true(1) means it is a Restricted Load
         Balancing type and setting it to false(2) means it is a
         General Load Balancing group type.

         This object should not be changed while its group ID is referenced
         by an active entry in docsLoadBalRestrictCmEntry."

    DEFVAL { false }
    ::= { docsLoadBalGrpEntry 2 }
```

```
docsLoadBalGrpInitTech OBJECT-TYPE
    SYNTAX      ChannelChgInitTechMap
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The initialization techniques that the CMTS can use when load
         balancing cable modems in the load balancing group.
         By default this object is initialized with all the defined
         bits having a value of '1'."
    ::= { docsLoadBalGrpEntry 3 }

docsLoadBalGrpDefaultPolicy OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Each Load Balancing Group has a default Load Balancing
         Policy.  A policy is described by a set of conditions
         (rules) that govern the load balancing process for
         a cable modem.  The CMTS assigns this Policy ID value to
         a cable modem associated with the group ID when the cable
         modem does not signal a Policy ID during registration.

         The Policy ID value is intended to be a numeric reference to
         a row entry in docsLoadBalPolicyEntry. However, It is not
         required to have an existing or active entry in
         docsLoadBalPolicyEntry when setting the value of
         docsLoadBalGrpDefaultPolicy, in which case it indicates
         no policy is associated with the load Balancing Group.

         The Policy ID of value 0 is reserved to indicate no policy is
         associated with the load balancing group."
    DEFVAL { 0 }
    ::= { docsLoadBalGrpEntry 4 }

docsLoadBalGrpEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Setting this object to true(1) enables internal
        autonomous load balancing on this group. Setting it to
        false(2) disables the load balancing operation on this
        group."
    DEFVAL { true }
    ::= { docsLoadBalGrpEntry 5 }

docsLoadBalGrpChgOverSuccess OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of successful load balancing change over
         operations initiated within this load balancing group."
    ::= { docsLoadBalGrpEntry 6 }

docsLoadBalGrpChgOverFails OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of failed load balancing change over operations
         initiated within this load balacing group."
    ::= { docsLoadBalGrpEntry 7 }
```

```
docsLoadBalGrpStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates the status of the row in this table.
         Setting this object to 'destroy' or 'notInService' for a group ID
         entry already referenced by docsLoadBalChannelEntry,
         docsLoadBalChnPairsEntry or docsLoadBalRestrictCmEntry returns
         an error code inconsistentValue."
    ::= { docsLoadBalGrpEntry 8 }


-- ---------------------------------------------------------------------
-- It contains all the upstream and downstream channels within the
-- load balancing group.
-- ---------------------------------------------------------------------

docsLoadBalChannelTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsLoadBalChannelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Lists all upstream and downstream channels associated with
         load balancing groups."
    ::= { docsLoadBalGrpObjects 2 }

docsLoadBalChannelEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalChannelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "Lists a specific upstream or downstream, within a
          load Balancing group.
          An entry in this table exists for each ifEntry with an ifType
          of docsCableDownstream(128) and docsCableUpstream(129)
          associated with the Load Balancing Group.
          Entries in this table persist after CMTS initialization."
    INDEX { docsLoadBalGrpId, docsLoadBalChannelIfIndex }
    ::= { docsLoadBalChannelTable 1 }

DocsLoadBalChannelEntry ::= SEQUENCE {
    docsLoadBalChannelIfIndex InterfaceIndex,
    docsLoadBalChannelStatus  RowStatus
    }

docsLoadBalChannelIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "The ifIndex of either the downstream or upstream."
::= { docsLoadBalChannelEntry 1 }


docsLoadBalChannelStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates the status of the rows in this table.
         Creating entries in this table requires an existing
         value for docsLoadBalGrpId in docsLoadBalGrpEntry and
         an existing value of docsLoadBalChannelIfIndex in
         ifEntry, otherwise is rejected with error 'noCreation'.
```

```
         Setting this object to 'destroy' or 'notInService for a
         a row entry that is being referenced by
         docsLoadBalChnPairsEntry is rejected with error code
         inconsistentValue."
::= { docsLoadBalChannelEntry 2 }

-- ---------------------------------------------------------------------
-- docsLoadBalChnPairsTable is used to override the initialization
-- techniques for specific channel pairs within a Load Balancing Group.
-- ---------------------------------------------------------------------

docsLoadBalChnPairsTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsLoadBalChnPairsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains pairs of upstream channels
         within a Load Balancing Group. Entries in this
         table are used to override the initialization techniques
         defined for the associated Load Balancing Group."
    ::= { docsLoadBalGrpObjects 3 }

docsLoadBalChnPairsEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalChnPairsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "An entry in this table describes a channel pair for which an
          initialization technique override is needed.
          On a CMTS which supports logical upstream channels
         (ifType is equal to docsCableUpstreamChannel(205)),
         the entries in this table correspond to pairs of ifType 205 .
         On a CTS which only supports physical upstream channels
         (iftype is equal to docsCableUpstream(129)), the entries in this
         table correspond to pairs of ifType 129.
         Entries in this table persist after CMTS initialization."
    INDEX { docsLoadBalGrpId, docsLoadBalChnPairsIfIndexDepart,
            docsLoadBalChnPairsIfIndexArrive }
    ::= { docsLoadBalChnPairsTable 1 }

DocsLoadBalChnPairsEntry ::= SEQUENCE {
    docsLoadBalChnPairsIfIndexDepart     InterfaceIndex,
    docsLoadBalChnPairsIfIndexArrive     InterfaceIndex,
    docsLoadBalChnPairsOperStatus   INTEGER,
    docsLoadBalChnPairsInitTech     ChannelChgInitTechMap,
    docsLoadBalChnPairsRowStatus    RowStatus
   }

docsLoadBalChnPairsIfIndexDepart OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This index indicates the ifIndex of the upstream channel from
         which a cable modem would depart in a load balancing channel
         change operation."
::= { docsLoadBalChnPairsEntry 1 }

docsLoadBalChnPairsIfIndexArrive OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This index indicates the ifIndex of the upstream channel on
         which a cable modem would arrive in a load balancing channel
```

```
        change operation."
::= { docsLoadBalChnPairsEntry 2 }

docsLoadBalChnPairsOperStatus OBJECT-TYPE
    SYNTAX      INTEGER {
          operational(1),
          notOperational(2)
      }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Operational status of the channel pair.  The value
         operational(1) indicates that ifOperStatus of both channels
         is up(1). The value notOperational(2) means that ifOperStatus
         of one or both is not up(1)."
::= { docsLoadBalChnPairsEntry 3 }


docsLoadBalChnPairsInitTech OBJECT-TYPE
    SYNTAX      ChannelChgInitTechMap
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
         "Specifies initialization technique for load balancing
          for the Depart/Arrive pair.
          By default this object's value is the initialization
          technique configured for the Load Balancing Group
          indicated by docsLoadBalGrpId."
::= { docsLoadBalChnPairsEntry 4 }


docsLoadBalChnPairsRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The object for conceptual rows creation.
         An attempt to create a row with values for
         docsLoadBalChnPairsIfIndexDepart or
         docsLoadBalChnPairsIfIndexArrive which are not a part
         of the Load Balancing Group (or for a 2.0 CMTS are not
         logical channels (ifType 205)) are rejected with a
         'noCreation' error status reported.
         There is no restriction on settings columns in this table
         when the value of docsLoadBalChnPairsRowStatus is active(1)."
::= { docsLoadBalChnPairsEntry 5 }

-- ---------------------------------------------------------------------
-- Restricted load balancing groups are defined to cater to a specific
-- group of modems to accomodate a topology specific or provisioning
-- specific restriction.
-- Restricted load balancing groups shall be configured with the CM MAC
-- addresses (or references to it). Such a group MUST apply only to the
-- modems configured in the group.
-- ---------------------------------------------------------------------

docsLoadBalRestrictCmTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsLoadBalRestrictCmEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Lists all cable modems in each Restricted Load Balancing
         Groups."
    ::= { docsLoadBalGrpObjects 4 }
```

```
docsLoadBalRestrictCmEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalRestrictCmEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry of  modem within a restricted load balancing
         group type.
         An entry represents a cable modem that is associated with the
         Restricted Load Balancing Group ID of a Restricted Load
         Balancing Group.
         Entries in this table persist after CMTS initialization."
    INDEX { docsLoadBalGrpId,
            docsLoadBalRestrictCmIndex }
    ::= { docsLoadBalRestrictCmTable 1 }

DocsLoadBalRestrictCmEntry ::= SEQUENCE {
    docsLoadBalRestrictCmIndex       Unsigned32,
    docsLoadBalRestrictCmMACAddr     MacAddress,
    docsLoadBalRestrictCmMacAddrMask OCTET STRING,
    docsLoadBalRestrictCmStatus      RowStatus
     }

docsLoadBalRestrictCmIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "The index that uniquely identifies an entry which
          represents restricted cable modem(s) within each Restricted
          Load Balancing Group."
::= { docsLoadBalRestrictCmEntry 1 }

docsLoadBalRestrictCmMACAddr OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Mac Address of the cable modem within the restricted load
         balancing group."
    ::= { docsLoadBalRestrictCmEntry 2 }

docsLoadBalRestrictCmMacAddrMask OBJECT-TYPE
    SYNTAX       OCTET STRING (SIZE (0 | 6))
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "A bit mask acting as a wildcard to associate a set
         of modem MAC addresses to the same Group ID.
         Cable modem lookup is performed first with entries
         containing this value not null, if several entries match,
         the largest consecutive bit match from MSB to LSB is used.
         Empty value is equivalent to the bit mask all in
         ones."
    DEFVAL { ''h }
    ::= { docsLoadBalRestrictCmEntry 3 }

docsLoadBalRestrictCmStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates the status of the rows in this table.
         The attempt to create an entry associated to a group ID with
         docsLoadBalGrpIsRestricted equal to false(2) returns an error
         'noCreation'.
```

```
         There is no restriction on settings columns in this table any
         time."
    ::= { docsLoadBalRestrictCmEntry 4 }

-- ---------------------------------------------------------------------
--
-- Load Balance policies allow control over the behavior of the autonomous
-- load balancing process on a per cable modem basis.  A load balancing
-- policy is described by a set of conditions/rules that govern the
-- autonomous load balancing process for the cable modem.
--
-- ---------------------------------------------------------------------

docsLoadBalPolicyTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsLoadBalPolicyEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table describes the set of Load Balancing policies.
         Rows in this table might be referenced by rows in
         docsLoadBalGrpEntry."
    ::= { docsLoadBalPolicyObjects 1 }

docsLoadBalPolicyEntry OBJECT-TYPE
    SYNTAX          DocsLoadBalPolicyEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Entries containing rules for policies.
         When a load balancing policy is defined by multiple
         rules, all the rules apply.

         Load balancing rules can be created to allow for
         specific vendor-defined load balancing actions.
         However there is a basic rule that the CMTS is
         required to support by configuring a pointer
         in docsLoadBalPolicyRulePtr to the table
         docsLoadBalBasicRuleTable. Vendor specific rules
         may be added by pointing the object
         docsLoadBalPolicyRulePtr to propietary mib structures.
         Entries in this table persist after CMTS initialization."
    INDEX { docsLoadBalPolicyId, docsLoadBalPolicyRuleId }
    ::= { docsLoadBalPolicyTable 1 }

DocsLoadBalPolicyEntry ::= SEQUENCE {
    docsLoadBalPolicyId         Unsigned32,
    docsLoadBalPolicyRuleId     Unsigned32,
    docsLoadBalPolicyRulePtr    RowPointer,
    docsLoadBalPolicyRowStatus  RowStatus
    }

docsLoadBalPolicyId   OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index identifying the Load Balancing Policy."
    ::= { docsLoadBalPolicyEntry 1 }

docsLoadBalPolicyRuleId   OBJECT-TYPE
    SYNTAX       Unsigned32 (1..4294967295)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An index for the rules entries associated within a policy."
```

```
    ::= { docsLoadBalPolicyEntry 2 }


docsLoadBalPolicyRulePtr OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
         "A pointer to an entry in a rule table. e.g
          docsLoadBalBasicRuleEnable in docsLoadBalBasicRuleEntry.
          A value pointing to zeroDotZero, an inactive Row or a
          non-existing entry is treated as no rule defined for this
          policy entry."
    DEFVAL {zeroDotZero }
    ::= { docsLoadBalPolicyEntry 3}

docsLoadBalPolicyRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of this conceptual row.
         There is no restriction on settings columns in this table
         when the value of docsLoadBalPolicyRowStatus is active(1).
         Setting this object to 'destroy' or 'notInService' for a row
         entry that is being referenced by docsLoadBalGrpDefaultPolicy in
         docsLoadBalGrpEntry returns an error code inconsistentValue."
    ::= { docsLoadBalPolicyEntry 5 }

-- ---------------------------------------------------------
-- docsLoadBalBasicRuleTable defines a DOCSIS required Policy
-- Ruleset for Load Balancing. A Policy ID may have multiple
-- rules, each rule pointing to ruleset structures like this
-- table or a vendor defined one.
-- ---------------------------------------------------------

docsLoadBalBasicRuleTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsLoadBalBasicRuleEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "DOCSIS defined basic ruleset for load Balancing Policy.
        This table enables of disable load balancing for the groups
        pointing to this ruleset in the policy group."
    ::= { docsLoadBalPolicyObjects 2 }

docsLoadBalBasicRuleEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalBasicRuleEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "An entry of DOCSIS defined basic ruleset.
        The object docsLoadBalBasicRuleEnable is used for
        instantiating an entry in this table via a RowPointer.
        Entries in this table persist after CMTS initialization."
    INDEX { docsLoadBalBasicRuleId }
    ::= { docsLoadBalBasicRuleTable 1 }

DocsLoadBalBasicRuleEntry ::= SEQUENCE {
    docsLoadBalBasicRuleId  Unsigned32,
    docsLoadBalBasicRuleEnable INTEGER,
    docsLoadBalBasicRuleDisStart Unsigned32,
    docsLoadBalBasicRuleDisPeriod Unsigned32,
    docsLoadBalBasicRuleRowStatus RowStatus
    }
```

```
docsLoadBalBasicRuleId   OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The unique index for this row."
    ::= { docsLoadBalBasicRuleEntry 1 }


docsLoadBalBasicRuleEnable   OBJECT-TYPE
    SYNTAX      INTEGER {
                enabled(1),
                disabled(2),
                disabledPeriod(3)
                }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "When using this ruleset, load balancing is enabled or disabled
         by the values enabled(1) and disabled(2) respectively.
         Additionally, a Load Balancing disabling period is defined in
         docsLoadBalBasicRuleDisStart and docsLoadBalBasicRuleDisPeriod
         if this object value is set to disabledPeriod(3)."
    ::= { docsLoadBalBasicRuleEntry 2 }


docsLoadBalBasicRuleDisStart   OBJECT-TYPE
    SYNTAX      Unsigned32 (0..86400)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "if object docsLoadBalBasicRuleEnable is disablePeriod(3)
         Load Balancing is disabled starting at this object value time
         (seconds from 12 AM). Otherwise, this object has no meaning."
    DEFVAL { 0 }
    ::= { docsLoadBalBasicRuleEntry 3 }

docsLoadBalBasicRuleDisPeriod   OBJECT-TYPE
    SYNTAX      Unsigned32 (0..86400)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If object docsLoadBalBasicRuleEnable is disablePeriod(3)
         Load Balancing is disabled for the period of time defined
         between docsLoadBalBasicRuleDisStart and
         docsLoadBalBasicRuleDisStart plus the period of time of
         docsLoadBalBasicRuleDisPeriod. Otherwise, this object value
         has no meaning."
    DEFVAL { 0 }
    ::= { docsLoadBalBasicRuleEntry 4 }

docsLoadBalBasicRuleRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object is to create or delete rows in
             this table.  There is no restriction for changing
             this row status or object's values in this table
             at any time."
    ::= { docsLoadBalBasicRuleEntry 5 }
```

```
-- ------------------------------------------------------
-- This table AUGMENTS the docsIfCmtsCmStatusTable to provide
-- the ability to associate the GroupId, PolicyId and Priority
-- to a modem.
-- Association of these attributes can also be done via the
-- cable modem config file.
-- ------------------------------------------------------

docsLoadBalCmtsCmStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsLoadBalCmtsCmStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The list contains the load balancing attributes
        associated with the cable modem. "
    ::= { docsLoadBalSystem 4 }

docsLoadBalCmtsCmStatusEntry OBJECT-TYPE
    SYNTAX      DocsLoadBalCmtsCmStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Additional objects for docsIfCmtsCmStatusTable entry
        that relate to load balancing "
    AUGMENTS { docsIfCmtsCmStatusEntry }
    ::= { docsLoadBalCmtsCmStatusTable 1 }

DocsLoadBalCmtsCmStatusEntry ::= SEQUENCE {
    docsLoadBalCmtsCmStatusGroupId  Unsigned32,
    docsLoadBalCmtsCmStatusPolicyId Unsigned32,
    docsLoadBalCmtsCmStatusPriority Unsigned32
    }

docsLoadBalCmtsCmStatusGroupId OBJECT-TYPE
    SYNTAX     Unsigned32
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The Group ID associated with this cable modem."
    ::= { docsLoadBalCmtsCmStatusEntry 1 }

docsLoadBalCmtsCmStatusPolicyId OBJECT-TYPE
    SYNTAX     Unsigned32
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The Policy ID associated with this cable modem."
    ::= { docsLoadBalCmtsCmStatusEntry 2 }

docsLoadBalCmtsCmStatusPriority OBJECT-TYPE
    SYNTAX     Unsigned32
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The Priority associated with this cable modem."
    ::= { docsLoadBalCmtsCmStatusEntry 3 }

-- ------------------------------------------------------
-- Conformance definitions
-- ------------------------------------------------------

docsLoadBalConformance OBJECT IDENTIFIER ::= { docsLoadBalanceMib 2 }
docsLoadBalCompliances OBJECT IDENTIFIER ::= { docsLoadBalConformance 1 }
docsLoadBalGroups      OBJECT IDENTIFIER ::= { docsLoadBalConformance 2 }
```

```
docsLoadBalBasicCompliance MODULE-COMPLIANCE
     STATUS current
     DESCRIPTION
          "The compliance statement for DOCSIS load balancing
           systems."

MODULE -- docsLoadBalancingMib
        MANDATORY-GROUPS {
                 docsLoadBalSystemGroup,
                 docsLoadBalParametersGroup,
                 docsLoadBalPoliciesGroup,
                 docsLoadBalBasicRuleGroup,
                 docsLoadBalCmtsCmStatusGroup
        }
::= { docsLoadBalCompliances 1}

docsLoadBalSystemGroup OBJECT-GROUP
       OBJECTS {
           docsLoadBalEnable,
           docsLoadBalChgOverMacAddress,
           docsLoadBalChgOverDownFrequency,
           docsLoadBalChgOverUpChannelId,
           docsLoadBalChgOverInitTech,
           docsLoadBalChgOverCmd,
           docsLoadBalChgOverCommit,
           docsLoadBalChgOverLastCommit,
           docsLoadBalChgOverStatusMacAddr,
           docsLoadBalChgOverStatusDownFreq,
           docsLoadBalChgOverStatusUpChnId,
           docsLoadBalChgOverStatusInitTech,
           docsLoadBalChgOverStatusCmd,
           docsLoadBalChgOverStatusValue,
           docsLoadBalChgOverStatusUpdate
       }
       STATUS  current
       DESCRIPTION
           "A collection of objects providing system-wide
            parameters for load balancing."
       ::= { docsLoadBalGroups 1}

docsLoadBalParametersGroup OBJECT-GROUP
       OBJECTS {
           docsLoadBalGrpIsRestricted,
           docsLoadBalGrpInitTech,
           docsLoadBalGrpDefaultPolicy,
           docsLoadBalGrpEnable,
           docsLoadBalGrpChgOverSuccess,
           docsLoadBalGrpChgOverFails,
           docsLoadBalGrpStatus,
           docsLoadBalChannelStatus,
           docsLoadBalChnPairsOperStatus,
           docsLoadBalChnPairsInitTech,
           docsLoadBalChnPairsRowStatus,
           docsLoadBalRestrictCmMACAddr,
           docsLoadBalRestrictCmMacAddrMask,
           docsLoadBalRestrictCmStatus
       }
       STATUS  current
       DESCRIPTION
           "A collection of objects containing the load balancing
            parameters."
       ::= { docsLoadBalGroups 2}
```

```
docsLoadBalPoliciesGroup OBJECT-GROUP
       OBJECTS {
           docsLoadBalPolicyRulePtr,
           docsLoadBalPolicyRowStatus
       }
       STATUS  current
       DESCRIPTION
           "A collection of objects providing policies."
       ::= { docsLoadBalGroups 3}

docsLoadBalBasicRuleGroup OBJECT-GROUP
       OBJECTS {
           docsLoadBalBasicRuleEnable,
           docsLoadBalBasicRuleDisStart,
           docsLoadBalBasicRuleDisPeriod,
           docsLoadBalBasicRuleRowStatus
       }
       STATUS current
       DESCRIPTION
            "DOCSIS defined basic Ruleset for load balancing
             policies."
       ::= { docsLoadBalGroups 4}

docsLoadBalCmtsCmStatusGroup OBJECT-GROUP
       OBJECTS {
           docsLoadBalCmtsCmStatusGroupId,
           docsLoadBalCmtsCmStatusPolicyId,
           docsLoadBalCmtsCmStatusPriority
       }
       STATUS current
       DESCRIPTION
            "Cable mode status extension objects."
       ::= { docsLoadBalGroups 5}
```

This page intentionally left blank.

Annex J Requirements for DOCS-QOS-MIB (mandatory)[1]

```
DOCS-QOS-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Integer32,
    Counter32,
    IpAddress,
    Unsigned32
      FROM SNMPv2-SMI
    TEXTUAL-CONVENTION,
    MacAddress,
    RowStatus,
    TruthValue,
    DisplayString,
    TimeStamp
      FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE
      FROM SNMPv2-CONF
    ifIndex,
    InterfaceIndex
      FROM IF-MIB
    docsIfMib
      FROM DOCS-IF-MIB;
docsQosMIB   MODULE-IDENTITY
    LAST-UPDATED    "0010180000Z" -- Oct 18, 2000
    ORGANIZATION    "IETF IPCDN Working Group"
    CONTACT-INFO
        "
         Co-Author: Michael Patrick
         Postal:    Motorola ISG
        20 Cabot Blvd, MS M4-30
        Mansfield, MA 02048-1193
        U.S.A.
         Phone:     +1 508 261 5707
         E-mail:    michael.patrick@motorola.com"
    DESCRIPTION
        "This is the management information for
         Quality Of Service (QOS) for DOCSIS 1.1."
    REVISION        "0010180000Z" -- October 18, 2000
    DESCRIPTION
        "Published as draft-ietf-ipcdn-qos-mib-04.txt.
        Changes from qos-mib-03 include:
        - Moved six objects from docsQosServiceFlowTable back
          to docsQosParamSetTable.
        - Added five counters to docsQosDynamicServiceStatsTable for
          DCC counts.
        - Removed notApplicable(256) from docsQosParamSetSchedulingType
        - Clarified reported values of docsQosParamSetTable objects.
          The CMTS reports any CMTS-specific default value it is
          using, and unknown or not applicable params are reported as zero.
        - Add docsQosPktClassBitMap
        - Add docsQosParamSetBitMap
        - Restore docsQosParamSetServiceClassName
        - Add 5 objects to docsQosServiceFlowLogTable
        - Add docsQosServiceClassDirection
        "
```

1. Added new Annex per ECN OSSSIv2.0-N-04.0126-6 by GO on 3/16/04.

```
    ::= { docsIfMib 7 }                -- BPIPlus mib is docsIfMIb 6
docsQosMIBObjects  OBJECT IDENTIFIER ::= { docsQosMIB 1 }

-- Textual Conventions
IfDirection ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        " Indicates a direction on an RF MAC interface.
        The value downstream(1) is from Cable Modem
        Termination System to Cable Modem.
        The value upstream(2) is from Cable Modem to
        Cable Modem Termination System."
    SYNTAX          INTEGER {
          downstream(1),
          upstream(2)
        }

BitRate ::= TEXTUAL-CONVENTION
    DISPLAY-HINT    "d"
    STATUS          current
    DESCRIPTION
        " The rate of traffic in unit of bits per second.
        Used to specify traffic rate for QOS."
    SYNTAX          Unsigned32
SchedulingType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        " The scheduling service provided by a CMTS for an
        upstream service flow. If the parameter is omitted
        from an upstream QOS Parameter Set, this object takes
        the value of bestEffort (2). This parameter must be
        reported as undefined (1) for downstream QOS Parameter
        Sets."
    SYNTAX          INTEGER {
         undefined (1),
         bestEffort (2),
         nonRealTimePollingService(3),
         realTimePollingService(4),
         unsolictedGrantServiceWithAD(5),
         unsolictedGrantService(6)
        }

--------------------------------------------------------------------------
--
-- Packet Classifier Table
--
docsQosPktClassTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosPktClassEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the packet classification
        configured on the CM or CMTS.
        The model is that a packet either received
        as input from an interface or transmitted
        for output on an interface may be compared
        against an ordered list of rules pertaining to
        the packet contents. Each rule is a row of this
        table. A matching rule provides a service flow
        id to to which the packet is classified.
        All rules need to match for a packet to match
        a classifier.
        The objects in this row correspond to a set of
        Classifier Encoding parameters in a DOCSIS
        MAC management message. The docsQosPktClassBitMap
```

```
        indicates which particular parameters were present
        in the classifier as signalled in the DOCSIS message.
        If the referenced parameter was not present
        in the signalled DOCSIS 1.1 Classifier, the
        corresponding object in this row reports a
        value as specified in the DESCRIPTION section.
        "
    ::= { docsQosMIBObjects 1 }

docsQosPktClassEntry OBJECT-TYPE
    SYNTAX          DocsQosPktClassEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " An entry in this table provides a single packet
        classifier rule. The index ifIndex is an ifType
        of docsCableMaclayer(127)."
    INDEX {
            ifIndex,
            docsQosServiceFlowId,
            docsQosPktClassId
          }

    ::= { docsQosPktClassTable 1 }

DocsQosPktClassEntry ::= SEQUENCE {
    docsQosPktClassId               Integer32,
    docsQosPktClassDirection        IfDirection,
    docsQosPktClassPriority         Integer32,
    docsQosPktClassIpTosLow         OCTET STRING,
    docsQosPktClassIpTosHigh        OCTET STRING,
    docsQosPktClassIpTosMask        OCTET STRING,
    docsQosPktClassIpProtocol       Integer32,
    docsQosPktClassIpSourceAddr     IpAddress,
    docsQosPktClassIpSourceMask     IpAddress,
    docsQosPktClassIpDestAddr       IpAddress,
    docsQosPktClassIpDestMask       IpAddress,
    docsQosPktClassSourcePortStart  Integer32,
    docsQosPktClassSourcePortEnd    Integer32,
    docsQosPktClassDestPortStart    Integer32,
    docsQosPktClassDestPortEnd      Integer32,
    docsQosPktClassDestMacAddr      MacAddress,
    docsQosPktClassDestMacMask      MacAddress,
    docsQosPktClassSourceMacAddr    MacAddress,
    docsQosPktClassEnetProtocolType INTEGER,
    docsQosPktClassEnetProtocol     Integer32,
    docsQosPktClassUserPriLow       Integer32,
    docsQosPktClassUserPriHigh      Integer32,
    docsQosPktClassVlanId           Integer32,
    docsQosPktClassState            INTEGER,
    docsQosPktClassPkts             Counter32,
    docsQosPktClassBitMap           BITS
  }

docsQosPktClassId       OBJECT-TYPE
    SYNTAX          Integer32 (1..65535)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Index assigned to packet classifier entry by
        the CMTS which is unique per service flow."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.3.2"
    ::= { docsQosPktClassEntry 1 }
```

```
docsQosPktClassDirection OBJECT-TYPE
    SYNTAX          IfDirection
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Indicates the direction to which the classifier
        is applied."
    ::= { docsQosPktClassEntry 2 }

docsQosPktClassPriority OBJECT-TYPE
    SYNTAX          Integer32 (0..255)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value specifies the order of evaluation
        of the classifiers.
        The higher the value the higher the priority.
        The value of 0 is used as default in
        provisioned service flows classifiers.
        The default value of 64 is used for dynamic
        service flow classifiers.
        If the referenced parameter is not present
        in a classifier, this object reports the default value
        as defined above."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.3.5"
    ::= { docsQosPktClassEntry 3 }

docsQosPktClassIpTosLow OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The low value of a range of TOS byte values.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 0."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.1"
    ::= { docsQosPktClassEntry 4 }

docsQosPktClassIpTosHigh OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The 8-bit high value of a range of TOS byte
        values.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 0."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.1"
    ::= { docsQosPktClassEntry 5 }

docsQosPktClassIpTosMask OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The mask value is bitwise ANDed with TOS byte
        in an IP packet and this value is used check
        range checking of TosLow and TosHigh.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 0."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.1"
    ::= { docsQosPktClassEntry 6 }
```

```
docsQosPktClassIpProtocol OBJECT-TYPE
    SYNTAX          Integer32 (0..258)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object indicates the value of the IP
        Protocol field required for IP packets to match
        this rule.
        The value 256 matches traffic with any IP Protocol
        value. The value 257 by convention matches both TCP
        and UDP.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 258."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.2"
    ::= { docsQosPktClassEntry 7 }

docsQosPktClassIpSourceAddr OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies the value of the IP
        Source Address required for packets to match
        this rule. An IP packet matches the rule when
        the packet ip source address bitwise ANDed
        with the docsQosPktClassIpSourceMask value
        equals the docsQosPktClassIpSourceAddr value.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        0.0.0.0."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.3"
    ::= { docsQosPktClassEntry 8 }

docsQosPktClassIpSourceMask OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies which bits of a packet's
        IP Source Address that are compared to match
        this rule.
        An IP packet matches the rule when the packet
        source address bitwise ANDed with the
        docsQosPktClassIpSourceMask value equals the
        docsQosIpPktClassSourceAddr value.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        0.0.0.0."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.4"
    ::= { docsQosPktClassEntry 9 }

docsQosPktClassIpDestAddr OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies the value of the IP
        Destination Address required for packets to
        match this rule. An IP packet matches the rule
        when the packet IP destination address
        bitwise ANDed with the
        docsQosPktClassIpDestMask value equals the
        docsQosPktClassIpDestAddr value.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
```

```
        0.0.0.0."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.5"
    ::= { docsQosPktClassEntry 10 }

docsQosPktClassIpDestMask OBJECT-TYPE
    SYNTAX          IpAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies which bits of a packet's
        IP Destination Address that are compared to
        match this rule.
        An IP packet matches the rule when the packet
        destination address bitwise ANDed with the
        docsQosPktClassIpDestMask value equals the
        docsQosPktClassIpDestAddr value.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        0.0.0.0."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.6"
    ::= { docsQosPktClassEntry 11}

docsQosPktClassSourcePortStart OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies the low end inclusive
        range of TCP/UDP source port numbers to which
        a packet is compared. This object is irrelevant
        for non-TCP/UDP IP packets.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 0."
    REFERENCE        "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.7"
    ::= { docsQosPktClassEntry 12 }

docsQosPktClassSourcePortEnd OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies the high end inclusive
        range of TCP/UDP source port numbers to which
        a packet is compared. This object is irrelevant
        for non-TCP/UDP IP packets.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        65535."
    REFERENCE        "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.9"
    ::= { docsQosPktClassEntry 13 }

docsQosPktClassDestPortStart OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies the low end inclusive
        range of TCP/UDP destination port numbers to
        which a packet is compared.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 0."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.9"
    ::= { docsQosPktClassEntry 14 }
```

```
docsQosPktClassDestPortEnd OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object specifies the high end inclusive
        range of TCP/UDP destination port numbers to which
        a packet is compared.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        65535."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.5.10"
    ::= { docsQosPktClassEntry 15 }

docsQosPktClassDestMacAddr OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " An Ethernet packet matches an entry when its
        destination MAC address bitwise ANDed with
        docsQosPktClassDestMacMask equals the value of
        docsQosPktClassDestMacAddr.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        '000000000000'H.
        "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.6.1"
    ::= { docsQosPktClassEntry 16 }

docsQosPktClassDestMacMask OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " An Ethernet packet matches an entry when its
        destination MAC address bitwise ANDed with
        docsQosPktClassDestMacMask equals the value of
        docsQosPktClassDestMacAddr.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        '000000000000'H.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.6.1"
    ::= { docsQosPktClassEntry 17 }

docsQosPktClassSourceMacAddr OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " An Ethernet packet matches this entry when its
        source MAC address equals the value of
        this object.
        If the referenced parameter is not present
        in a classifier, this object reports the value of
        'FFFFFFFFFFFF'H.
        "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.6.2"
    ::= { docsQosPktClassEntry 18 }
```

```
docsQosPktClassEnetProtocolType OBJECT-TYPE
    SYNTAX          INTEGER {
         none(0),
         ethertype(1),
         dsap(2),
         mac(3),
         all(4)
        }

    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object indicates the format of the layer 3
        protocol id in the Ethernet packet. A value of
        none(0) means that the rule does not use the
        layer 3 protocol type as a matching criteria.
        A value of ethertype(1) means that the rule
        applies only to frames which contains an
        EtherType value. Ethertype values are contained
        in packets using the Dec-Intel-Xerox (DIX)
        encapsulation or the RFC1042 Sub-Network Access
        Protocol (SNAP) encapsulation formats.
        A value of dsap(2) means that the rule applies
        only to frames using the IEEE802.3
        encapsulation format with a Destination Service
        Access Point (DSAP) other
        than 0xAA (which is reserved for SNAP).
        A value of mac(3) means that the rule applies
        only to MAC management messages for MAC management
        messages.
        A value of all(4) means that the rule matches
        all Ethernet packets.
        If the Ethernet frame contains an 802.1P/Q Tag
        header (i.e. EtherType 0x8100), this object
        applies to the embedded EtherType field within
        the 802.1P/Q header.
        If the referenced parameter is not present
        in a classifier, this object reports the value of 0.
        "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.6.3"
    ::= { docsQosPktClassEntry 19 }

docsQosPktClassEnetProtocol OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " If docsQosEthPktClassProtocolType is none(0),
        this object is ignored when considering whether
        a packet matches the current rule.
        If dosQosPktClassEnetProtocolType is ethertype(1),
        this object gives the 16-bit value of the
        EtherType that the packet must match in order to
        match the rule.
        If docsQosPktClassEnetProtocolType is dsap(2), the
        lower 8 bits of this object's value must match the
        DSAP byte of the packet in order to match the
        rule.
        If docsQosPktClassEnetProtocolType is mac(3), the
        lower 8 bits of this object value represent a
        lower bound (inclusive) of MAC management message
        type codes matched, and the upper 8 bits of this
        object value represent the upper bound (inclusive)
        of matched MAC message type codes.  Certain
        message type codes are excluded from matching, as
```

```
        specified in the reference.
        If the Ethernet frame contains an 802.1P/Q Tag header
        (i.e. EtherType 0x8100), this object applies to the
        embedded EtherType field within the 802.1P/Q header.
        If the referenced parameter is not present in the
        classifier, the value of this object is reported as 0.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.6.3"
    ::= { docsQosPktClassEntry 20 }

-- docsQosPktClassUserPriApplies { docsQosPktClassEntry 21 }

--   was removed in revision -03.
docsQosPktClassUserPriLow OBJECT-TYPE
    SYNTAX          Integer32 (0..7)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object applies only to Ethernet frames
        using the 802.1P/Q tag header (indicated with
        EtherType 0x8100). Such frames include a 16-bit
        Tag that contains a 3 bit Priority field and
        a 12 bit VLAN number.
        Tagged Ethernet packets must have a 3-bit
        Priority field within the range of
        docsQosPktClassPriLow and docsQosPktClassPriHigh in
        order to match this rule.
        If the referenced parameter is not present in the
        classifier, the value of this object is reported as 0.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.7.1"
    ::= { docsQosPktClassEntry 22 }

docsQosPktClassUserPriHigh OBJECT-TYPE
    SYNTAX          Integer32 (0..7)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object applies only to Ethernet frames
        using the 802.1P/Qtag header (indicated with
        EtherType 0x8100). Such frames include a 16-bit
        Tag that contains a 3 bit Priority field and
        a 12 bit VLAN number.
        Tagged Ethernet packets must have a 3-bit
        Priority field within the range of
        docsQosPktClassPriLow and
        docsQosPktClassPriHigh in order to match this
        rule.
        If the referenced parameter is not present in the
        classifier, the value of this object is reported
        as 7.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.7.1"
    ::= { docsQosPktClassEntry 23 }

docsQosPktClassVlanId OBJECT-TYPE
    SYNTAX          Integer32 (0..4095)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object applies only to Ethernet frames
        using the 802.1P/Q tag header.
        If this object's value is nonzero, tagged
        packets must have a VLAN Identifier that matches
        the value in order to match the rule.
```

```
        Only the least significant 12 bits of this object's
        value are valid.
        If the referenced parameter is not present in the
        classifier, the value of this object is reported
        as 0.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.7.2"
    ::= { docsQosPktClassEntry 24 }

docsQosPktClassState OBJECT-TYPE
    SYNTAX          INTEGER {
         active(1),
         inactive(2)
        }

    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object indicates whether or not the classifier
        is enabled to classify packets to a Service Flow.
        If the referenced parameter is not present in the
        classifier, the value of this object is reported
        as active(1).
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.3.6"
    ::= { docsQosPktClassEntry 25 }

docsQosPktClassPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object counts the number of packets that have
        been classified using this entry."
    ::= { docsQosPktClassEntry 26 }

docsQosPktClassBitMap OBJECT-TYPE
    SYNTAX          BITS {              -- Reference SP-RFIv1.1-I05-000714
           rulePriority(0),     -- Appendix C.2.1.3.4
           activationState(1),  -- Appendix C.2.1.3.6
           ipTos(2),            -- Appendix C.2.1.5.1
           ipProtocol(3),       -- Appendix C.2.1.5.2
           ipSourceAddr(4),     -- Appendix C.2.1.5.3
           ipSourceMask(5),     -- Appendix C.2.1.5.4
           ipDestAddr(6),       -- Appendix C.2.1.5.5
           ipDestMask(7),       -- Appendix C.2.1.5.6
           sourcePortStart(8),  -- Appendix C.2.1.5.7
           sourcePortEnd(9),    -- Appendix C.2.1.5.8
           destPortStart(10),   -- Appendix C.2.1.5.9
           destPortEnd(11),     -- Appendix C.2.1.5.10
           destMac(12),         -- Appendix C.2.1.6.1
           sourceMac(13),       -- Appendix C.2.1.6.2
           ethertype(14),       -- Appendix C.2.1.6.3
           userPri(15),         -- Appendix C.2.1.7.1
           vlanId(16)           -- Appendix C.2.1.7.2
        }

    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object indicates which parameter encodings
        were actually present in the DOCSIS packet classifier
        encoding signalled in the DOCSIS message that
        created the classifier.
        A bit of of this object is set to 1 if the parameter
```

```
        indicated by the comment was present in the classifier
        encoding, and 0 otherwise.
        Note that BITS are encoded most significant bit
        first, so that if e.g. bits 6 and 7 are set, this object
        is encoded as the octet string '030000'H.
                    "
    ::= { docsQosPktClassEntry 27 }

--
-- QOS Parameter Set Table
--
docsQosParamSetTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosParamSetEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the set of DOCSIS 1.1 QOS
        parameters defined in a managed device.
        The ifIndex index specifies a DOCSIS MAC Domain.
        The docsQosServiceFlowId index specifies a particular
        Service Flow.
        The docsQosParamSetType index indicates whether
        the active, admitted, or provisioned QOS Parameter
        Set is being described by the row.
        Only the QOS Parameter Sets of Docsis 1.1 service
        flows are represented in this table.  Docsis 1.0
        QOS service profiles are not represented in this
        table.
        Each row corresponds to a DOCSIS QOS Parameter Set
        as signaled via DOCSIS MAC management messages.
        Each object in the row corresponds to one or
        part of one DOCSIS 1.1 Service Flow Encoding.
        The docsQosParamSetBitMap object in the row indicates
        which particular parameters were signalled in
        the original registration or dynamic service
        request message that created the QOS Parameter Set.
        In many cases, even if a QOS Parameter Set parameter
        was not signalled, the DOCSIS specification calls
        for a default value to be used. That default value
        is reported as the value of the corresponding object
        in this row.
        Many objects are not applicable depending on
        the service flow direction or upstream scheduling
        type.  The object value reported in this case
        is specified in the DESCRIPTION clause.
        "
    ::= { docsQosMIBObjects 2 }

docsQosParamSetEntry OBJECT-TYPE
    SYNTAX          DocsQosParamSetEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
    "A unique set of QOS parameters."
    INDEX {
        ifIndex, docsQosServiceFlowId, docsQosParamSetType
          }
    ::= { docsQosParamSetTable 2 }

-- Type of docsQosParamSet Entry { docsQosParamSetTable 1 } was
--   changed with revision -03
DocsQosParamSetEntry ::= SEQUENCE {
    docsQosParamSetServiceClassName   DisplayString,
    docsQosParamSetPriority           Integer32,
    docsQosParamSetMaxTrafficRate     BitRate,
```

```
    docsQosParamSetMaxTrafficBurst    Unsigned32,
    docsQosParamSetMinReservedRate    BitRate,
    docsQosParamSetMinReservedPkt     Integer32,
    docsQosParamSetActiveTimeout      Integer32,
    docsQosParamSetAdmittedTimeout    Integer32,
    docsQosParamSetMaxConcatBurst     Integer32,
    docsQosParamSetSchedulingType     SchedulingType,
    docsQosParamSetNomPollInterval    Unsigned32,
    docsQosParamSetTolPollJitter      Unsigned32,
    docsQosParamSetUnsolicitGrantSize Integer32,
    docsQosParamSetNomGrantInterval   Unsigned32,
    docsQosParamSetTolGrantJitter     Unsigned32,
    docsQosParamSetGrantsPerInterval  Integer32,
    docsQosParamSetTosAndMask         OCTET STRING,
    docsQosParamSetTosOrMask          OCTET STRING,
    docsQosParamSetMaxLatency         Unsigned32,
    docsQosParamSetType               INTEGER,
    docsQosParamSetRequestPolicyOct   OCTET STRING,
    docsQosParamSetBitMap             BITS
    }

-- Removed docsQosParamSetRowType { docsQosParamSetEntry 1  }

--    with revision -03
-- Removed docsQosParamSetIndex { docsQosParamSetEntry 2 }

--    with revision -03
-- Removed docsQosParamSetRowStatus { docsQosParamSetEntry 3}

--    with revision -03
docsQosParamSetServiceClassName OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Refers to the Service Class Name that the
        parameter set values were derived.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object is a zero length string.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.3.4"
    ::= { docsQosParamSetEntry 4 }

docsQosParamSetPriority OBJECT-TYPE
    SYNTAX          Integer32 (0..7)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The relative priority of a service flow.
        Higher numbers indicate higher priority.
        This priority should only be used to differentiate
        service flow with identical parameter sets.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object is 0.  If the parameter is
        not applicable, the reported value is 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.2"
    ::= { docsQosParamSetEntry 5 }

docsQosParamSetMaxTrafficRate OBJECT-TYPE
    SYNTAX          BitRate
    MAX-ACCESS      read-only
    STATUS          current
```

```
    DESCRIPTION
        " Maximum sustained traffic rate allowed for this
        service flow in bits/sec. Must count all MAC frame
        data PDU from the bytes following the MAC header HCS to
        the end of the CRC. The number of bytes
        forwarded is limited during any time interval.
        The value 0 means no maximum traffic rate is
        enforced. This object applies to both upstream and
        downstream service flows.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object is 0. If the parameter is
        not applicable, it is reported as 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.3"
    ::= { docsQosParamSetEntry 6 }

docsQosParamSetMaxTrafficBurst OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the token bucket size in bytes
        for this parameter set. The value is calculated
        from the byte following the MAC header HCS to
        the end of the CRC. This object is applied in
        conjunction with docsQosParamSetMaxTrafficRate to
        calculate maximum sustained traffic rate.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object for scheduling types
        bestEffort (2), nonRealTimePollingService(3),
        and realTimePollingService(4) is 3044.
        If this parameter is not applicable, it is reported
        as 0."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.4"
    ::= { docsQosParamSetEntry 7 }

docsQosParamSetMinReservedRate OBJECT-TYPE
    SYNTAX          BitRate
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the guaranteed minimum rate in
        bits/sec for this parameter set. The value is
        calculated from the byte following the MAC
        header HCS to the end of the CRC. The default
        value of 0 has the meaning that no bandwidth
        is reserved.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object is 0. If the parameter
        is not applicable, it is reported as 0.
        "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.5"
    ::= { docsQosParamSetEntry 8 }

docsQosParamSetMinReservedPkt OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies an assumed minimum packet size in
        bytes for which the docsQosParamSetMinReservedRate
        will be provided. The value is calculated from
```

```
        the byte following the MAC header HCS to the
        end of the CRC.
        If the referenced parameter is omitted from a
        DOCSIS QOS parameter set, the default value is
        CMTS implementation dependent. In this case, the
        CMTS reports the default value it is using and the
        CM reports a value of 0. If the referenced
        parameter is not applicable to the direction or
        scheduling type of the service flow, both CMTS and
        CM report this object's value as 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.6"
    ::= { docsQosParamSetEntry 9 }

docsQosParamSetActiveTimeout OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the maximum duration in seconds that
        resources remain unused on an active service
        flow before CMTS signals that both active and
        admitted parameters set are null.
        The default value of 0 signifies an
        infinite amount of time.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object is 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.7"
    ::= { docsQosParamSetEntry 10 }

docsQosParamSetAdmittedTimeout OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the maximum duration in seconds that
        resources remain in admitted state before
        resources must be released.
        The value of 0 signifies an infinite amount
        of time.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set, the
        default value of this object is 200.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.5.8"
    DEFVAL          { 200 }

    ::= { docsQosParamSetEntry 11 }

docsQosParamSetMaxConcatBurst OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the maximum concatenated burst in
        bytes which an upstream  service flow is allowed.
        The value is calculated from the FC byte of the
        Concatenation MAC Header to the last CRC byte in
        of the last concatenated MAC frame, inclusive.
        The value of 0 specifies no maximum burst.
        If the referenced parameter is not present in the
```

```
        corresponding DOCSIS QOS Parameter Set, the default
        value of this object is 1522. If the parameter is
        not applicable, this object's value is reported
        as 0."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.1"
    ::= { docsQosParamSetEntry 12 }

docsQosParamSetSchedulingType OBJECT-TYPE
    SYNTAX          SchedulingType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the upstream scheduling service used for
        upstream service flow.
        If the referenced parameter is not present in the
        corresponding DOCSIS QOS Parameter Set of an
        upstream service flow, the default value of this
        object is bestEffort(2). For QOS parameter sets of
        downstream service flows, this object's value is
        reported as undefined(1).
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.2"
    ::= { docsQosParamSetEntry 13 }

--  Changed type of  docsQosParamSetRequestPolicy { docsQosParamSetEntry 14 }

--    to docsQosParamSetRequestPolicyOct { docsQosParamSetEntry 25 }

docsQosParamSetNomPollInterval OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the nominal interval in microseconds
        between successive unicast request
        opportunities on an upstream service flow.
        This object applies only to upstream service flows
        with schedulingType of value
        nonRealTimePollingService(3),
        realTimePollingService(4), and
        unsolictedGrantServiceWithAD(5).  The parameter is
        mandatory for realTimePollingService(4).  If the
        parameter is omitted with
        nonRealTimePollingService(3), the CMTS uses an
        implementation dependent value.  If the parameter
        is omitted with unsolictedGrantServiceWithAD(5),
        the CMTS uses as a default value the value of the
        Nominal Grant Interval parameter.  In all cases,
        the CMTS reports the value it is using when the
        parameter is applicable.  The CM reports the
        signaled parameter value if it was signaled,
        and 0 otherwise.
        If the referenced parameter is not applicable to
        the direction or scheduling type of the
        corresponding DOCSIS QOS Parameter Set, both
        CMTS and CM report this object's value as 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.4"
    ::= { docsQosParamSetEntry 15 }

docsQosParamSetTolPollJitter OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-only
```

```
    STATUS          current
    DESCRIPTION
        " Specifies the maximum amount of time in
        microseconds that the unicast request interval
        may be delayed from the nominal periodic
        schedule on an upstream service flow.
        This parameter is applicable only to upstream
        service flows with a Schedulingtype of
        realTimePollingService(4) or
        unsolictedGrantServiceWithAD(5).
        If the referenced parameter is applicable but not
        present in the corresponding DOCSIS QOS Parameter
        Set, the CMTS uses an implementation dependent
        value and reports the value it is using.
        The CM reports a value of 0 in this case.
        If the parameter is not applicable to the
        direction or upstream scheduling type of the
        service flow, both CMTS and CM report this
        object's value as 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.5"
    ::= { docsQosParamSetEntry 16 }

docsQosParamSetUnsolicitGrantSize OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the unsolicited grant size in bytes.
        The grant size includes the entired MAC frame
        data PDU from the Frame Control byte to end of
        the MAC frame.
        The referenced parameter is applicable only
        for upstream flows with a SchedulingType of
        of unsolicitedGrantServicewithAD(5) or
        unsolicitedGrantService(6), and is mandatory
        when applicable. Both CMTS and CM report
        the signaled value of the parameter in this
        case.
        If the referenced parameter is not applicable to
        the direction or scheduling type of the
        corresponding DOCSIS QOS Parameter Set, both
        CMTS and CM report this object's value as 0.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.6"
    ::= { docsQosParamSetEntry 17 }

docsQosParamSetNomGrantInterval OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the nominal interval in microseconds
        between successive data grant opportunities
        on an upstream service flow.
        The referenced parameter is applicable only
        for upstream flows with a SchedulingType of
        of unsolicitedGrantServicewithAD(5) or
        unsolicitedGrantService(6), and is mandatory
        when applicable. Both CMTS and CM report the
        signaled value of the parameter in this case.
        If the referenced parameter is not applicable to
        the direction or scheduling type of the
        corresponding DOCSIS QOS Parameter Set, both
```

```
        CMTS and CM report this object's value as 0.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.7"
    ::= { docsQosParamSetEntry 18 }

docsQosParamSetTolGrantJitter OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the maximum amount of time in
        microseconds that the transmission opportunities
        may be delayed from the nominal periodic schedule.
        The referenced parameter is applicable only
        for upstream flows with a SchedulingType of
        of unsolicitedGrantServicewithAD(5) or
        unsolicitedGrantService(6), and is mandatory
        when applicable. Both CMTS and CM report the
        signaled value of the parameter in this case.
        If the referenced parameter is not applicable to
        the direction or scheduling type of the
        corresponding DOCSIS QOS Parameter Set, both
        CMTS and CM report this object's value as 0.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.8"
    ::= { docsQosParamSetEntry 19 }

docsQosParamSetGrantsPerInterval OBJECT-TYPE
    SYNTAX          Integer32 (0..127)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the number of data grants per Nominal
        Grant Interval
        (docsQosParamSetNomGrantInterval).
        The referenced parameter is applicable only
        for upstream flows with a SchedulingType of
        of unsolicitedGrantServicewithAD(5) or
        unsolicitedGrantService(6), and is mandatory
        when applicable. Both CMTS and CM report the
        signaled value of the parameter in this case.
        If the referenced parameter is not applicable to
        the direction or scheduling type of the
        corresponding DOCSIS QOS Parameter Set, both
        CMTS and CM report this object's value as 0.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.9"
    ::= { docsQosParamSetEntry 20 }

docsQosParamSetTosAndMask OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the AND mask for IP TOS byte for overwriting
        IP packets TOS value.  The IP packets TOS byte is
        bitwise ANDed with docsQosParamSetTosAndMask and
        result is bitwise ORed with docsQosParamSetTosORMask
        and result is written to IP packet TOS byte.
        A value of 'FF'H for docsQosParamSetTosAndMask and
        a value of '00'H for docsQosParamSetTosOrMask means
        that IP Packet TOS byte is not overwritten.
        This combination is reported if the referenced
        parameter is not present in a QOS Parameter Set."
```

```
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.10"
    ::= { docsQosParamSetEntry 21 }

docsQosParamSetTosOrMask OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the OR mask for IP TOS byte.
        See the description of docsQosParamSetTosAndMask
        for further details."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.10"
    ::= { docsQosParamSetEntry 22 }

docsQosParamSetMaxLatency OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies the maximum latency between the
        reception of a packet by the CMTS on its NSI
        and the forwarding of the packet to the RF
        interface. A value of 0 signifies no maximum
        latency enforced. This object only applies to
        downstream service flows.
        If the referenced parameter is not present in the
        corresponding downstream DOCSIS QOS Parameter Set,
        the default value is 0. This parameter is
        not applicable to upstream DOCSIS QOS Parameter Sets,
        and its value is reported as 0 in this case.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.7.1"
    ::= { docsQosParamSetEntry 23 }

docsQosParamSetType     OBJECT-TYPE
    SYNTAX          INTEGER {
          active (1),
          admitted (2),
          provisioned (3)
        }

    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Defines the type of the QOS parameter set defined
        by this row. active(1) indicates the Active QOS
        parameter set, describing the service currently
        being provided by the Docsis MAC domain to the
        service flow. admitted(2) indicates the Admitted
        QOS Parameter Set, describing services reserved by
        by the Docsis MAC domain for use by the service flow.
        provisioned (3) describes the QOS Parameter Set
        defined in the DOCSIS CM Configuration file for
        the service flow."
    REFERENCE      "SP-RFIv1.1-I05-000714, 8.1.5"
    ::= { docsQosParamSetEntry 24 }

docsQosParamSetRequestPolicyOct OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(4))
        -- A 32-bit mask represented most significant byte
        -- first. The 32 bit integer represented in this manner
        -- equals the binary value of the referenced integer
        -- parameter of the DOCSIS RFI specification.
        -- The BITS syntax is not used in order to avoid
```

```
        -- the confusion caused by different bit numbering
        -- conventions.
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Specifies which transmit interval opportunities
        the CM omits for upstream transmission requests and
        packet transmissions. This object takes its
        default value for downstream service flows.
        Unless otherwise indicated, a bit value of 1 means
        that a CM must *not* use that opportunity for
        upstream transmission.
        Calling bit 0 the least significant bit of the
        least significant (4th) octet, and increasing
        bit number with significance, the bit definitions
        are as defined below:
        broadcastReqOpp(0):
            all CMs broadcast request opportunities
        priorityReqMulticastReq(1):
            priority request multicast request opportunities
        reqDataForReq(3):
            request/data opportunities for requests
        reqDataForData(4):
            request/data opportunities for data
        concatenateData(5):
            concatenate data
        fragmentData(6):
            fragment data
        suppresspayloadheaders(7):
            suppress payload headers
        dropPktsExceedUGSize(8):
            A value of 1 mean that service flow must drop
            packet that do not fit in the Unsolicited
            Grant size
        If the referenced parameter is not present in
        a QOS Parameter Set, the value of this object is
        reported as '00000000'H.
        "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.6.3"
    ::= { docsQosParamSetEntry 25 }

docsQosParamSetBitMap OBJECT-TYPE
                   -- Each bit corresponds to a parameter
                   -- from SP-RFI-v1.1-I05-000714, Appendix C
    SYNTAX          BITS {      -- in the indicated section number.
           trafficPriority(0),     -- C.2.2.5.2
           maxTrafficRate(1),      -- C.2.2.5.3
           maxTrafficBurst(2),     -- C.2.2.5.4
           minReservedRate(3),     -- C.2.2.5.5
           minReservedPkt(4),      -- C.2.2.5.6
           activeTimeout(5),       -- C.2.2.5.7
           admittedTimeout(6),     -- C.2.2.5.8
           maxConcatBurst(7),      -- C.2.2.6.1
           schedulingType(8),      -- C.2.2.6.2
           requestPolicy(9),       -- C.2.2.6.3
           nomPollInterval(10),    -- C.2.2.6.4
           tolPollJitter(11),      -- C.2.2.6.5
           unsolicitGrantSize(12), -- C.2.2.6.6
           nomGrantInterval(13),   -- C.2.2.6.7
           tolGrantJitter(14),     -- C.2.2.6.8
           grantsPerInterval(15),  -- C.2.2.6.9
           tosOverwrite(16),       -- C.2.2.6.10
           maxLatency(17)          -- C.2.2.7.1
        }
```

```
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " This object indicates the set of QOS Parameter
        Set parameters actually signaled in the
        DOCSIS registration or dynamic service request
        message that created the QOS Parameter Set.
        A bit is set to 1 when the parameter described
        by the indicated reference section is present
        in the original request.
        Note that when Service Class names are expanded,
        the registration or dynamic response message may
        contain parameters as expanded by the CMTS based
        on a stored service class. These expanded
        parameters are *not* indicated by a 1 bit in this
        object.
        Note that even though some QOS Parameter Set
        parameters may not be signalled in a message
        (so that the paramater's bit in this object is 0)
        the DOCSIS specification calls for default
        values to be used. These default values are
        reported as the corresponding object's value in
        the row.
        Note that BITS objects are encoded most
        significant bit first. For example, if bits
        1 and 16 are set, the value of this object
        is the octet string '400080'H.
                    "
::= { docsQosParamSetEntry 26 }

--
--  Service Flow Table
--
docsQosServiceFlowTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceFlowEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the set of Docsis-QOS
        Service Flows in a managed device. "
    ::= { docsQosMIBObjects 3 }

docsQosServiceFlowEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceFlowEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Describes a service flow.
        An entry in the table exists for each
        Service Flow ID. The ifIndex is an
        ifType of docsCableMaclayer(127)."
    INDEX {
            ifIndex,
            docsQosServiceFlowId
          }

    ::= { docsQosServiceFlowTable 1 }

DocsQosServiceFlowEntry ::= SEQUENCE {
    docsQosServiceFlowId                    Unsigned32,
    docsQosServiceFlowSID                   Unsigned32,
    docsQosServiceFlowDirection             IfDirection,
    docsQosServiceFlowPrimary               TruthValue
    }
```

```
docsQosServiceFlowId    OBJECT-TYPE
    SYNTAX          Unsigned32 (1..4294967295)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " An index assigned to a service flow by CMTS."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.3.2"
    ::= { docsQosServiceFlowEntry 1 }

-- Remove docsQosServiceFlowProvisionedParamSetIndex
--    {docsQosServiceFlowEntry 2} with revision -03
-- Remove docsQosServiceFlowAdmittedParamSetIndex
--    {docsQosServiceFlowEntry 3} with revision -03
-- Remove docsQosServiceFlowActiveParamSetIndex
--    {docsQosServiceFlowEntry 4} with revision -03
docsQosServiceFlowSID  OBJECT-TYPE
    SYNTAX          Unsigned32 (0..16383)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Service Identifier (SID) assigned to an
        admitted or active service flow. This object
        reports a value of 0 if a Service Id is not
        associated with the service flow. Only active
        or admitted upstream service flows will have a
        Service Id (SID)."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.3.3"
    ::= { docsQosServiceFlowEntry 6 }

docsQosServiceFlowDirection OBJECT-TYPE
    SYNTAX          IfDirection
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The direction of the service flow."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.1/2"
    ::= { docsQosServiceFlowEntry 7 }

docsQosServiceFlowPrimary OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Object reflects whether service flow is the primary
        or a secondary service flow.
        A primary service flow is the default service flow
        for otherwise unclassified traffic and all MAC
        messages."
    REFERENCE      "SP-RFIv1.1-I05-000714, Section 8.1 "
    ::= { docsQosServiceFlowEntry 8 }

-- Moved docsQosServiceFlow'ActiveTimeout, 'AdmittedTimeout,
-- 'SchedulingType, 'RequestPolicy, 'TosAndMask, and 'TosOrMask
-- back to docsQosParamSetTable with QOS-MIB-04.
--
--  Service Flow Stats Table
--
docsQosServiceFlowStatsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceFlowStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes statistics associated with the
        Service Flows in a managed device. "
    ::= { docsQosMIBObjects 4 }
```

```
docsQosServiceFlowStatsEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceFlowStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Describes a set of service flow statistics.
        An entry in the table exists for each
        Service Flow ID. The ifIndex is an
        ifType of docsCableMaclayer(127)."
    INDEX {
            ifIndex,
            docsQosServiceFlowId
          }

    ::= { docsQosServiceFlowStatsTable 1 }

DocsQosServiceFlowStatsEntry ::= SEQUENCE {
    docsQosServiceFlowPkts                      Counter32,
    docsQosServiceFlowOctets                    Counter32,
    docsQosServiceFlowTimeCreated               TimeStamp,
    docsQosServiceFlowTimeActive                Counter32,
    docsQosServiceFlowPHSUnknowns               Counter32,
    docsQosServiceFlowPolicedDropPkts           Counter32,
    docsQosServiceFlowPolicedDelayPkts          Counter32
    }

docsQosServiceFlowPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of packet counted on this service flow."
    ::= { docsQosServiceFlowStatsEntry 1 }

docsQosServiceFlowOctets OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of octets counted on this service flow
        after payload header suppression."
    ::= { docsQosServiceFlowStatsEntry 2 }

docsQosServiceFlowTimeCreated OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value of sysUpTime when the service flow
        was created."
    ::= { docsQosServiceFlowStatsEntry 3 }

docsQosServiceFlowTimeActive OBJECT-TYPE
    SYNTAX          Counter32
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The total time that service flow has been active."
    ::= { docsQosServiceFlowStatsEntry 4 }

docsQosServiceFlowPHSUnknowns OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
```

```
    STATUS          current
    DESCRIPTION
        " The number of packet with unknown payload header
        suppression index."
    ::= { docsQosServiceFlowStatsEntry 5 }

docsQosServiceFlowPolicedDropPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of packets dropped due to policing of
        the service flow, especially to limit the maximum
        rate of the flow."
    ::= { docsQosServiceFlowStatsEntry 6 }

docsQosServiceFlowPolicedDelayPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of packet delayed due to policing of
        the service flow, especially to limit the maximum
        rate of the flow."
    ::= { docsQosServiceFlowStatsEntry 7 }

--
--  Upstream Service Flow Stats Table (CMTS ONLY)
--
docsQosUpstreamStatsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosUpstreamStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes statistics associated with
        upstream service flows. All counted frames must
        be received without an FCS error."
    ::= { docsQosMIBObjects 5 }

docsQosUpstreamStatsEntry OBJECT-TYPE
    SYNTAX          DocsQosUpstreamStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Describes a set of upstream service flow statistics.
        An entry in the table exists for each
        upstream Service Flow in a managed device.
        The ifIndex is an ifType of docsCableMaclayer(127)."
    INDEX {
            ifIndex,
            docsQosSID
          }

    ::= { docsQosUpstreamStatsTable 1 }

DocsQosUpstreamStatsEntry ::= SEQUENCE {
    docsQosSID                          Integer32,
    docsQosUpstreamFragments            Counter32,
    docsQosUpstreamFragDiscards         Counter32,
    docsQosUpstreamConcatBursts         Counter32
    }

docsQosSID OBJECT-TYPE
    SYNTAX          Integer32 (1..16383)
    MAX-ACCESS      not-accessible
```

```
    STATUS          current
    DESCRIPTION
        " Identifies a service id for an admitted or active
        upstream service flow."
    ::= { docsQosUpstreamStatsEntry 1 }

-- Renamed in revision -03 from docsQosUpstreamFragPkts
docsQosUpstreamFragments OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of fragmentation headers received on an
        upstream  service flow, regardless of whether
        the fragment was correctly reassembled into a
        valid packet. "
    ::= { docsQosUpstreamStatsEntry 2 }

docsQosUpstreamFragDiscards OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of upstream fragments discarded and not
        assembled into a valid upstream packet."
    ::= { docsQosUpstreamStatsEntry 3 }

docsQosUpstreamConcatBursts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of concatenation headers received on an
        upstream service flow."
    ::= { docsQosUpstreamStatsEntry 4 }

--
--  Dynamic Service Stats Table
--
docsQosDynamicServiceStatsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosDynamicServiceStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes statistics associated with the
        Dynamic Service Flows in a managed device. "
    ::= { docsQosMIBObjects 6 }

docsQosDynamicServiceStatsEntry OBJECT-TYPE
    SYNTAX          DocsQosDynamicServiceStatsEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Describes a set of dynamic service flow statistics.
        Two entries exist for each Docsis mac layer
        interface for the upstream and downstream direction.
        On the CMTS, the downstream direction row indicates
        messages transmitted or transactions originated
        by the CMTS. The upstream direction row indicates
        messages received or transaction originated by the
        CM. On the CM, the downstream direction row
        indicates messages received or transactions
        originated by the CMTS. The upstream direction
        row indicates messages transmitted by the CM or
        transactions originated by the CM.
```

```
        The ifIndex is an ifType of docsCableMaclayer(127)."
    INDEX {
            ifIndex,
            docsQosIfDirection
          }

    ::= { docsQosDynamicServiceStatsTable 1 }

DocsQosDynamicServiceStatsEntry ::= SEQUENCE {
    docsQosIfDirection                          IfDirection,
    docsQosDSAReqs                              Counter32,
    docsQosDSARsps                              Counter32,
    docsQosDSAAcks                              Counter32,
    docsQosDSCReqs                              Counter32,
    docsQosDSCRsps                              Counter32,
    docsQosDSCAcks                              Counter32,
    docsQosDSDReqs                              Counter32,
    docsQosDSDRsps                              Counter32,
    docsQosDynamicAdds                          Counter32,
    docsQosDynamicAddFails                      Counter32,
    docsQosDynamicChanges                       Counter32,
    docsQosDynamicChangeFails                   Counter32,
    docsQosDynamicDeletes                       Counter32,
    docsQosDynamicDeleteFails                   Counter32,
    docsQosDCCReqs                              Counter32,
    docsQosDCCRsps                              Counter32,
    docsQosDCCAcks                              Counter32,
    docsQosDCCs                                 Counter32,
    docsQosDCCFails                             Counter32,
    docsQosDCCRspDeparts                        Counter32,
    docsQosDCCRspArrives                        Counter32

   }

docsQosIfDirection OBJECT-TYPE
    SYNTAX          IfDirection
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " The direction of interface."
    ::= { docsQosDynamicServiceStatsEntry 1 }

docsQosDSAReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Addition Requests"
    ::= { docsQosDynamicServiceStatsEntry 2 }

docsQosDSARsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Addition Responses"
    ::= { docsQosDynamicServiceStatsEntry 3 }

docsQosDSAAcks OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Addition Acknowledgements."
    ::= { docsQosDynamicServiceStatsEntry 4 }
```

```
docsQosDSCReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Change Requests"
    ::= { docsQosDynamicServiceStatsEntry 5 }

docsQosDSCRsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Change Responses"
    ::= { docsQosDynamicServiceStatsEntry 6 }

docsQosDSCAcks OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Change Acknowlegements."
    ::= { docsQosDynamicServiceStatsEntry 7 }

docsQosDSDReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Delete Requests"
    ::= { docsQosDynamicServiceStatsEntry 8 }

docsQosDSDRsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Service Delete Responses"
    ::= { docsQosDynamicServiceStatsEntry 9 }

docsQosDynamicAdds OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of successful Dynamic Service Addition
        transactions."
    ::= { docsQosDynamicServiceStatsEntry 10 }

docsQosDynamicAddFails OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of failed Dynamic Service Addition
        transactions."
    ::= { docsQosDynamicServiceStatsEntry 11 }
```

```
docsQosDynamicChanges OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of successful Dynamic Service Change
        transactions."
    ::= { docsQosDynamicServiceStatsEntry 12 }

docsQosDynamicChangeFails OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of failed Dynamic Service Change
        transactions."
    ::= { docsQosDynamicServiceStatsEntry 13 }

docsQosDynamicDeletes OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of successful Dynamic Service Delete
        transactions."
    ::= { docsQosDynamicServiceStatsEntry 14 }

docsQosDynamicDeleteFails OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of failed Dynamic Service Delete
        transactions."
    ::= { docsQosDynamicServiceStatsEntry 15 }

docsQosDCCReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Channel Change Request messages
        traversing an interface. This count is nonzero only on
        downstream direction rows."
    ::= { docsQosDynamicServiceStatsEntry 16 }

docsQosDCCRsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Channel Change Response messages
        traversing an interface. This count is only counted on upstream
        direction rows. This count should include number of retries. This
    includes all types of Dynamic Channel Change Response messages."
    ::= { docsQosDynamicServiceStatsEntry 17 }
```

```
docsQosDCCAcks OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of Dynamic Channel Change Acknowledgement
        messages traversing an interface. This count is nonzero only
        on downstream direction rows."
    ::= { docsQosDynamicServiceStatsEntry 18 }

docsQosDCCs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of successful Dynamic Channel Change
        transactions. This count is only counted on downstream direction
        rows.
           For CMTS the following rules apply:
          - If either the initialization technique 0 is utilized or a
            CM is moved to a different CMTS chassis, the DCCtransaction
            is successful when the DCC-RSP (depart) message is received
            from the CM.
          - In all the other cases, the DCC transaction is successfulif
            either the DCC-RSP (arrive) message from the CM is received
            on the new channel or the presence of the CM on the new   channel
            is internally confirmed.
          - The docsQosDCCs is only incremented on the DOCSIS MAC layer
            interface where the DCC was originated."
    REFERENCE "SP-RFIv2.0-I04-037030, Figures 11-59, 11-60, 11-61, 11-62"
    ::= { docsQosDynamicServiceStatsEntry 19 }

docsQosDCCFails OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of failed Dynamic Channel Change
        transactions. This count is only counted on downstream direction
        rows.
        For CMTS, if the result of the Dynamic Channel Change is different
        from what is described in docsQosDCCs, the DCC transaction is a
        failure. The docsQosDCCFails is only incremented on theDOCSIS
        mac layer interface where the DCC was originated."
    REFERENCE "SP-RFIv2.0-I04-037030, Figures 11-59, 11-60, 11-61, 11-62"
    ::= { docsQosDynamicServiceStatsEntry 20 }

docsQosDCCRspDeparts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of Dynamic Channel Change Response (depart)
        messages traversing an interface. This count is only counted on
        upstream direction rows."
    REFERENCE "SP-RFIv2.0-I04-037030, Figures 11-59, 11-60, 11-61, 11-62"
    ::= { docsQosDynamicServiceStatsEntry 21 }
```

```
docsQosDCCRspArrives OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of Dynamic Channel Change Response (arrive)
        messages traversing an interface. This count is only counted on
        upstream direction rows. This count should include number of retries."
  REFERENCE "SP-RFIv2.0-I04-037030, Figures 11-59, 11-60, 11-61, 11-62"
  ::= { docsQosDynamicServiceStatsEntry 22 }

--
--  Service Flow Log Table (CMTS ONLY)
--
docsQosServiceFlowLogTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceFlowLogEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table contains a log of the disconnected
        Service Flows in a managed device."
    ::= { docsQosMIBObjects 7 }

docsQosServiceFlowLogEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceFlowLogEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " The information regarding a single disconnected
        service flow."
    INDEX {
            docsQosServiceFlowLogIndex
          }

    ::= { docsQosServiceFlowLogTable 1 }

DocsQosServiceFlowLogEntry ::= SEQUENCE {
    docsQosServiceFlowLogIndex                Unsigned32,
    docsQosServiceFlowLogIfIndex              InterfaceIndex,
    docsQosServiceFlowLogSFID                 Unsigned32,
    docsQosServiceFlowLogCmMac                MacAddress,
    docsQosServiceFlowLogPkts                 Counter32,
    docsQosServiceFlowLogOctets               Counter32,
    docsQosServiceFlowLogTimeDeleted          TimeStamp,
    docsQosServiceFlowLogTimeCreated          TimeStamp,
    docsQosServiceFlowLogTimeActive           Counter32,
    docsQosServiceFlowLogDirection            IfDirection,
    docsQosServiceFlowLogPrimary              TruthValue,
    docsQosServiceFlowLogServiceClassName     DisplayString,
    docsQosServiceFlowLogPolicedDropPkts      Counter32,
    docsQosServiceFlowLogPolicedDelayPkts     Counter32,
    docsQosServiceFlowLogControl              INTEGER
    }

docsQosServiceFlowLogIndex OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Unique index for a logged service flow."
    ::= { docsQosServiceFlowLogEntry 1 }
```

```
docsQosServiceFlowLogIfIndex OBJECT-TYPE
    SYNTAX          InterfaceIndex
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The ifIndex of ifType docsCableMacLayter(127)
        on the CMTS where the service flow was present."
    ::= {  docsQosServiceFlowLogEntry 2 }

docsQosServiceFlowLogSFID    OBJECT-TYPE
    SYNTAX          Unsigned32 (1..4294967295)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The index assigned to the service flow by the CMTS."
    ::= {  docsQosServiceFlowLogEntry 3 }

docsQosServiceFlowLogCmMac OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The MAC address for the cable modem associated with
        the service flow."
    ::= { docsQosServiceFlowLogEntry 4 }

docsQosServiceFlowLogPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of packets counted on this service flow
        after payload header suppression."
    ::= { docsQosServiceFlowLogEntry 5 }

docsQosServiceFlowLogOctets OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The number of octets counted on this service flow
        after payload header suppression."
    ::= { docsQosServiceFlowLogEntry 6 }

docsQosServiceFlowLogTimeDeleted OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value of sysUpTime when the service flow
        was deleted."
    ::= { docsQosServiceFlowLogEntry 7 }

docsQosServiceFlowLogTimeCreated OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value of sysUpTime when the service flow
        was created."
    ::= { docsQosServiceFlowLogEntry 8 }
```

```
docsQosServiceFlowLogTimeActive OBJECT-TYPE
    SYNTAX          Counter32
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The total time that service flow was active."
    ::= { docsQosServiceFlowLogEntry 9 }

-- docsQosServiceFlowLogControl was formerly { docsQosServiceFlowLogEntry 10}

-- and was renumbered in version -04.
docsQosServiceFlowLogDirection OBJECT-TYPE
    SYNTAX          IfDirection
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value of docsQosServiceFlowDirection
        for the service flow."
    ::= { docsQosServiceFlowLogEntry  11}

docsQosServiceFlowLogPrimary OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value of docsQosServiceFlowPrimary for the
        service flow."
    ::= { docsQosServiceFlowLogEntry  12}

docsQosServiceFlowLogServiceClassName OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The value of docsQosParamSetServiceClassName for
        the provisioned QOS Parameter Set of the
        service flow."
    ::= { docsQosServiceFlowLogEntry  13}

docsQosServiceFlowLogPolicedDropPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The final value of docsQosServiceFlowPolicedDropPkts
        for the service flow."
    ::= { docsQosServiceFlowLogEntry  14}

docsQosServiceFlowLogPolicedDelayPkts OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The final value of docsQosServiceFlowPolicedDelayPkts
        for the service flow."
    ::= { docsQosServiceFlowLogEntry  15}
```

```
docsQosServiceFlowLogControl OBJECT-TYPE
    SYNTAX          INTEGER {
        active(1),
        destroy(6)
        }

    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        " Setting this object to the value destroy(6) removes
        this entry from the table.
        Reading this object return the value active(1)."
    ::= { docsQosServiceFlowLogEntry 16}

--
-- Service Class Table (CMTS ONLY)
--
docsQosServiceClassTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceClassEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the set of Docsis-QOS
        Service Classes in a CMTS. "
    ::= { docsQosMIBObjects 8 }

docsQosServiceClassEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceClassEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " A provisioned service class on a CMTS.
                Each entry defines a template for certain
                DOCSIS QOS Parameter Set values. When a CM
                creates or modifies an Admitted QOS Parameter Set for a
                Service Flow, it may reference a Service Class
                Name instead of providing explicit QOS Parameter
                Set values. In this case, the CMTS populates
                the QOS Parameter Set with the applicable
                corresponding values from the named Service Class.
                Subsequent changes to a Service Class row do *not*
                affect the QOS Parameter Set values of any service flows
                already admitted.
                A service class template applies to only
                a single direction, as indicated in the
                docsQosServiceClassDirection object.
                "
    INDEX {
             docsQosServiceClassName
          }

    ::= { docsQosServiceClassTable 1 }

DocsQosServiceClassEntry ::= SEQUENCE {
    docsQosServiceClassName               DisplayString,
    docsQosServiceClassStatus             RowStatus,
    docsQosServiceClassPriority           Integer32,
    docsQosServiceClassMaxTrafficRate     BitRate,
    docsQosServiceClassMaxTrafficBurst    Unsigned32,
    docsQosServiceClassMinReservedRate    BitRate,
    docsQosServiceClassMinReservedPkt     Integer32,
    docsQosServiceClassMaxConcatBurst     Integer32,
    docsQosServiceClassNomPollInterval    Unsigned32,
    docsQosServiceClassTolPollJitter      Unsigned32,
    docsQosServiceClassUnsolicitGrantSize Integer32,
```

```
    docsQosServiceClassNomGrantInterval   Unsigned32,
    docsQosServiceClassTolGrantJitter     Unsigned32,
    docsQosServiceClassGrantsPerInterval  Integer32,
    docsQosServiceClassMaxLatency         Unsigned32,
    docsQosServiceClassActiveTimeout      Integer32,
    docsQosServiceClassAdmittedTimeout    Integer32,
    docsQosServiceClassSchedulingType     SchedulingType,
    docsQosServiceClassRequestPolicy      OCTET STRING,
    docsQosServiceClassTosAndMask         OCTET STRING,
    docsQosServiceClassTosOrMask          OCTET STRING,
    docsQosServiceClassDirection          IfDirection
    }

docsQosServiceClassName OBJECT-TYPE
    SYNTAX          DisplayString (SIZE(1..15))
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Service Class Name. DOCSIS specifies that the
        maximum size is 15 printable ASCII characters with
        a terminating zero. The terminating zero is not
        represented in this DisplayString syntax object.
                "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.3.4"
    ::= { docsQosServiceClassEntry 1 }

-- docsQosServiceClassParamSetIndex { docsQosServiceClassEntry 2 }

--    was removed in revision -03
docsQosServiceClassStatus OBJECT-TYPE
    SYNTAX          RowStatus
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Used to create or delete rows in this table."
    ::= { docsQosServiceClassEntry 3 }

docsQosServiceClassPriority OBJECT-TYPE
    SYNTAX          Integer32 (0..7)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetPriority."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 4 }

docsQosServiceClassMaxTrafficRate OBJECT-TYPE
    SYNTAX          BitRate
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetMaxTrafficRate."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 5 }

docsQosServiceClassMaxTrafficBurst OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetMaxTrafficBurst."
    DEFVAL          { 3044 }
```

```
    ::= { docsQosServiceClassEntry 6 }

docsQosServiceClassMinReservedRate OBJECT-TYPE
    SYNTAX          BitRate
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSEtMinReservedRate."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 7 }

docsQosServiceClassMinReservedPkt OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetMinReservedPkt."
    ::= { docsQosServiceClassEntry 8 }

docsQosServiceClassMaxConcatBurst OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetMaxConcatBurst."
    DEFVAL          { 1522 }

    ::= { docsQosServiceClassEntry 9 }

docsQosServiceClassNomPollInterval OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetNomPollInterval."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 10 }

docsQosServiceClassTolPollJitter OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetTolPollJitter."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 11 }

docsQosServiceClassUnsolicitGrantSize OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetUnsolicitGrantSize."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 12 }
```

```
docsQosServiceClassNomGrantInterval OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetNomGrantInterval."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 13 }

docsQosServiceClassTolGrantJitter OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetTolGrantJitter."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 14 }

docsQosServiceClassGrantsPerInterval OBJECT-TYPE
    SYNTAX          Integer32 (0..127)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetGrantsPerInterval."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 15 }

docsQosServiceClassMaxLatency OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "microseconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosParamSetClassMaxLatency."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.2.7.1"
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 16 }

docsQosServiceClassActiveTimeout OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    UNITS           "seconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosServiceFlowActiveTimeout."
    DEFVAL          { 0 }

    ::= { docsQosServiceClassEntry 17 }

docsQosServiceClassAdmittedTimeout OBJECT-TYPE
    SYNTAX          Integer32 (0..65535)
    UNITS           "seconds"
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosServiceFlowAdmittedTimeout."
    DEFVAL          { 200 }

    ::= { docsQosServiceClassEntry 18 }
```

```
docsQosServiceClassSchedulingType OBJECT-TYPE
    SYNTAX          SchedulingType
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosServiceFlowSchedulingType."
    DEFVAL          { bestEffort }

    ::= { docsQosServiceClassEntry 19 }

docsQosServiceClassRequestPolicy OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(4))
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosServiceFlowRequestPolicy."
    DEFVAL          { '00000000'H } -- no bits are set
    ::= { docsQosServiceClassEntry 20 }

docsQosServiceClassTosAndMask OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosServiceFlowTosAndMask."
    DEFVAL          { 'FF'H }

    ::= { docsQosServiceClassEntry 21 }

docsQosServiceClassTosOrMask OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(1))
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Template for docsQosServiceFlowTosOrMask."
    DEFVAL          { '00'H }

    ::= { docsQosServiceClassEntry 22 }

docsQosServiceClassDirection OBJECT-TYPE
    SYNTAX          IfDirection
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Specifies whether the service class template
        applies to upstream or downstream service flows."
    DEFVAL          { upstream }

    ::= { docsQosServiceClassEntry 23 }

--
-- Service Class PolicyTable
--
docsQosServiceClassPolicyTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceClassPolicyEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the set of Docsis-QOS
        Service Class Policies.
        This table is an adjunct to the
        docsDevFilterPolicy table.  Entries in
        docsDevFilterPolicy table can  point to
        specific rows in this table.
```

```
        This table permits mapping a packet to a service
        class name of an active service flow so long as
        a classifier does not exist at a higher
        priority.
                    "
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix E.2.1"
    ::= { docsQosMIBObjects 9 }

docsQosServiceClassPolicyEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceClassPolicyEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " A service class name policy entry."
    INDEX {
            docsQosServiceClassPolicyIndex
          }

    ::= { docsQosServiceClassPolicyTable 1 }

DocsQosServiceClassPolicyEntry ::= SEQUENCE {
    docsQosServiceClassPolicyIndex        Integer32,
    docsQosServiceClassPolicyName         DisplayString,
    docsQosServiceClassPolicyRulePriority Integer32,
    docsQosServiceClassPolicyStatus       RowStatus
    }

docsQosServiceClassPolicyIndex OBJECT-TYPE
    SYNTAX          Integer32 (1..2147483647)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " Index value to uniquely identify an entry in
        this table."
    ::= { docsQosServiceClassPolicyEntry 1 }

docsQosServiceClassPolicyName OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Service Class Name to identify the name of the
        service class flow to which the packet should be
        directed."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix E.2.1"
    ::= { docsQosServiceClassPolicyEntry 2 }

docsQosServiceClassPolicyRulePriority OBJECT-TYPE
    SYNTAX          Integer32 (0..255)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Service Class Policy rule priority for the
        entry."
    REFERENCE      "SP-RFIv1.1-I05-000714, Appendix C.2.1.3.5"
    ::= { docsQosServiceClassPolicyEntry 3 }

docsQosServiceClassPolicyStatus OBJECT-TYPE
    SYNTAX          RowStatus
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " Used to create or delete rows in this table.
        This object should not be deleted if it is
        reference by an entry in docsDevFilterPolicy.
```

```
        The reference should be deleted first."
    ::= { docsQosServiceClassPolicyEntry 4 }

--
-- Payload Header Suppression(PHS) Table
--
docsQosPHSTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosPHSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes set of payload header
        suppression entries."
    ::= { docsQosMIBObjects 10 }

docsQosPHSEntry OBJECT-TYPE
    SYNTAX          DocsQosPHSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " A payload header suppression entry.
        The ifIndex is an ifType of docsCableMaclayer(127).
        The index docsQosServiceFlowId selects one
        service flow from the cable MAC layer interface.
        The docsQosPktClassId index matches an
        index of the docsQosPktClassTable.
        "
    INDEX {
            ifIndex,
            docsQosServiceFlowId,
            docsQosPktClassId
          }

    ::= { docsQosPHSTable 1 }

DocsQosPHSEntry ::= SEQUENCE {
    docsQosPHSField            OCTET STRING,
    docsQosPHSMask             OCTET STRING,
    docsQosPHSSize             Integer32,
    docsQosPHSVerify           TruthValue,
    docsQosPHSIndex            Integer32
    }

-- The object docsQosPHSIndex used as an index {docsQosPHSEntry 1}

--    was changed to be a non-index column in revision -03.
docsQosPHSField         OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(0..255))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Payload header suppression field defines the
        bytes of the header which must be
        suppressed/restored by the sending/receiving
        device.
        The number of octets in this object should be
        the same as the value of docsQosPHSSize."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.2.10.1"
    ::= { docsQosPHSEntry 2 }
```

```
docsQosPHSMask          OBJECT-TYPE
    SYNTAX          OCTET STRING(SIZE(0..32))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Payload header suppression mask defines the
        bit mask which used in combination with the
        docsQosPHSField defines which bytes in header
        must be suppressed/restored by the sending or
        receiving device.
        Each bit of this bit mask corresponds to a byte
        in the docsQosPHSField, with the least
        significant  bit corresponding to first byte of
        the docsQosPHSField.
        Each bit of the bit mask specifies whether of
        not the corresponding byte should be suppressed
        in the packet. A bit value of '1' indicates that
        the byte should be suppressed by the sending
        device and restored by the receiving device.
        A bit value of '0' indicates that
        the byte should not be suppressed by the sending
        device or restored by the receiving device.
        If the bit mask does not contain a bit for each
        byte in the docsQosPHSField then the bit mask is
        extended with bit values of '1' to be the
        necessary length."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.2.10.3"
    ::= { docsQosPHSEntry 3 }

docsQosPHSSize          OBJECT-TYPE
    SYNTAX          Integer32 (0..255)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Payload header suppression size specifies the
        number of bytes in the header to be suppressed
        and restored.
        The value of this object must match the number
        of bytes in the docsQosPHSField."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.2.10.4"
    ::= { docsQosPHSEntry 4 }

docsQosPHSVerify       OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Payload header suppression verification value of
        'true' the sender must verify docsQosPHSField
        is the same as what is contained in the packet
        to be suppressed."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.2.10.5"
    ::= { docsQosPHSEntry 5 }

-- Removed dosQosPHSClassifierIndex {docsQosPHSEntry 6}

--   in revision -03.
docsQosPHSIndex         OBJECT-TYPE
    SYNTAX          Integer32 (1..255)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " Payload header suppression index uniquely
        references the PHS rule for a given service flow."
    REFERENCE       "SP-RFIv1.1-I05-000714, Appendix C.2.2.10.2"
```

```
    ::= { docsQosPHSEntry 7 }

--
-- docsQosCmtsMacToSrvFlowTable (CMTS Only)
--
docsQosCmtsMacToSrvFlowTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosCmtsMacToSrvFlowEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table provide for referencing the service flows
        associated with a particular cable modem. This allows
        for indexing into other docsQos tables that are
        indexed by docsQosServiceFlowId and ifIndex."
    ::= { docsQosMIBObjects 11 }

docsQosCmtsMacToSrvFlowEntry OBJECT-TYPE
    SYNTAX          DocsQosCmtsMacToSrvFlowEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " An entry is created by CMTS for each service flow
        connected to this CMTS."
    INDEX {
            docsQosCmtsCmMac,
            docsQosCmtsServiceFlowId
          }

    ::= { docsQosCmtsMacToSrvFlowTable 1 }

DocsQosCmtsMacToSrvFlowEntry ::= SEQUENCE {
    docsQosCmtsCmMac                MacAddress,
    docsQosCmtsServiceFlowId        Unsigned32,
    docsQosCmtsIfIndex              InterfaceIndex
    }

docsQosCmtsCmMac OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " The MAC address for the referenced CM."
    ::= { docsQosCmtsMacToSrvFlowEntry 1 }

docsQosCmtsServiceFlowId OBJECT-TYPE
    SYNTAX          Unsigned32 (1..4294967295)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " An index assigned to a service flow by CMTS."
    ::= { docsQosCmtsMacToSrvFlowEntry 2 }

docsQosCmtsIfIndex OBJECT-TYPE
    SYNTAX          InterfaceIndex
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        " The ifIndex of ifType docsCableMacLayter(127)
        on the CMTS that is connected to the Cable Modem."
    ::= { docsQosCmtsMacToSrvFlowEntry 3 }

--
-- Placeholder for notifications/traps.
--
```

```
docsQosNotification OBJECT IDENTIFIER   ::= { docsQosMIB 2 }

--
-- Conformance definitions
--
docsQosConformance  OBJECT IDENTIFIER   ::= { docsQosMIB 3 }

docsQosGroups       OBJECT IDENTIFIER   ::= { docsQosConformance 1 }

docsQosCompliances  OBJECT IDENTIFIER   ::= { docsQosConformance 2 }

docsQosCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
        "The compliance statement for MCNS Cable Modems and
         Cable Modem Termination Systems that implement DOCSIS
         Service Flows."
    MODULE  -- docsQosMIB
        MANDATORY-GROUPS { docsQosBaseGroup }

        GROUP docsQosCmtsGroup
        DESCRIPTION
            "This group is mandatory for only Cable Modem Termination
             Systems (CMTS) and not implemented for Cable Modems."
        GROUP docsQosParamSetGroup
        DESCRIPTION
            "This group is mandatory for Cable Modem Termination
             Systems (CMTS) and Cable Modems. Cable modems only implement
             objects in this group as read-only."
        GROUP docsQosSrvClassPolicyGroup
        DESCRIPTION
            "This group is optional for Cable Modem Termination
             Systems (CMTS) and Cable Modems. This group only needs to
             be implement if policy based service flow classification
             is implemented. See docsDevPolicyTable in
             DOCS-CABLE-DEVICE-MIB for more details. "
        GROUP docsQosServiceClassGroup
        DESCRIPTION
            "The docsQosServiceClassTable group of objects. "
        OBJECT  docsQosPktClassPkts
        DESCRIPTION
            "This object only needs to be implemented in entries
             that are classifying packets and not policing packets."
    ::= { docsQosCompliances 1 }

docsQosBaseGroup OBJECT-GROUP
    OBJECTS {
    docsQosPktClassDirection,
    docsQosPktClassPriority,
    docsQosPktClassIpTosLow,
    docsQosPktClassIpTosHigh,
    docsQosPktClassIpTosMask,
    docsQosPktClassIpProtocol,
    docsQosPktClassIpSourceAddr,
    docsQosPktClassIpSourceMask,
    docsQosPktClassIpDestAddr,
    docsQosPktClassIpDestMask,
    docsQosPktClassSourcePortStart,
    docsQosPktClassSourcePortEnd,
    docsQosPktClassDestPortStart,
    docsQosPktClassDestPortEnd,
    docsQosPktClassDestMacAddr,
    docsQosPktClassDestMacMask,
    docsQosPktClassSourceMacAddr,
    docsQosPktClassEnetProtocolType,
```

```
    docsQosPktClassEnetProtocol,
    docsQosPktClassUserPriLow,
    docsQosPktClassUserPriHigh,
    docsQosPktClassVlanId,
    docsQosPktClassState,
    docsQosPktClassPkts,
    docsQosPktClassBitMap,
    docsQosServiceFlowSID,
    docsQosServiceFlowDirection,
    docsQosServiceFlowPrimary,
    docsQosServiceFlowPkts,   -- not sure if CM should implement
    docsQosServiceFlowOctets,
    docsQosServiceFlowTimeCreated,
    docsQosServiceFlowTimeActive,
    docsQosServiceFlowPHSUnknowns,
    docsQosServiceFlowPolicedDropPkts,
    docsQosServiceFlowPolicedDelayPkts,
    docsQosDSAReqs,
    docsQosDSARsps,
    docsQosDSAAcks,
    docsQosDSCReqs,
    docsQosDSCRsps,
    docsQosDSCAcks,
    docsQosDSDReqs,
    docsQosDSDRsps,
    docsQosDynamicAdds,
    docsQosDynamicAddFails,
    docsQosDynamicChanges,
    docsQosDynamicChangeFails,
    docsQosDynamicDeletes,
    docsQosDynamicDeleteFails,
    docsQosDCCReqs,
    docsQosDCCRsps,
    docsQosDCCAcks,
    docsQosDCCs,
    docsQosDCCFails,
    docsQosDCCRspDeparts,
    docsQosDCCRspArrives,
    docsQosPHSField,
    docsQosPHSMask,
    docsQosPHSSize,
    docsQosPHSVerify,
    docsQosPHSIndex
    }

    STATUS  current
    DESCRIPTION
        "Group of objects implemented in both Cable Modems and
         Cable Modem Termination Systems."
    ::= { docsQosGroups 1 }

docsQosParamSetGroup OBJECT-GROUP
    OBJECTS {
    docsQosParamSetServiceClassName,
    docsQosParamSetPriority,
    docsQosParamSetMaxTrafficRate,
    docsQosParamSetMaxTrafficBurst,
    docsQosParamSetMinReservedRate,
    docsQosParamSetMinReservedPkt,
    docsQosParamSetActiveTimeout,
    docsQosParamSetAdmittedTimeout,
    docsQosParamSetMaxConcatBurst,
    docsQosParamSetSchedulingType,
    docsQosParamSetNomPollInterval,
    docsQosParamSetTolPollJitter,
```

```
    docsQosParamSetUnsolicitGrantSize,
    docsQosParamSetNomGrantInterval,
    docsQosParamSetTolGrantJitter,
    docsQosParamSetGrantsPerInterval,
    docsQosParamSetTosAndMask,
    docsQosParamSetTosOrMask,
    docsQosParamSetMaxLatency,
    docsQosParamSetRequestPolicyOct,
    docsQosParamSetBitMap
    }

    STATUS  current
    DESCRIPTION
        "Group of objects implemenent in both Cable Modems and
         Cable Modem Termination Systems for QOS parameter sets."
    ::= { docsQosGroups 2 }

docsQosCmtsGroup OBJECT-GROUP
    OBJECTS {
    docsQosUpstreamFragments,
    docsQosUpstreamFragDiscards,
    docsQosUpstreamConcatBursts,
    docsQosServiceFlowLogIfIndex,
    docsQosServiceFlowLogSFID,
    docsQosServiceFlowLogCmMac,
    docsQosServiceFlowLogPkts,
    docsQosServiceFlowLogOctets,
    docsQosServiceFlowLogTimeDeleted,
    docsQosServiceFlowLogTimeCreated,
    docsQosServiceFlowLogTimeActive,
    docsQosServiceFlowLogDirection,
    docsQosServiceFlowLogPrimary,
    docsQosServiceFlowLogServiceClassName,
    docsQosServiceFlowLogPolicedDropPkts,
    docsQosServiceFlowLogPolicedDelayPkts,
    docsQosServiceFlowLogControl,
    docsQosCmtsIfIndex        -- docsQosCmtsMacToSrvFlowTable required
    }

    STATUS  current
    DESCRIPTION
        "Mandatory group of objects implemented only in the CMTS."
    ::= { docsQosGroups 3 }

docsQosSrvClassPolicyGroup OBJECT-GROUP
    OBJECTS {
    docsQosServiceClassPolicyName,
    docsQosServiceClassPolicyRulePriority,
    docsQosServiceClassPolicyStatus
    }

    STATUS  current
    DESCRIPTION
        "Group of objects implemented in both Cable Modems and
         Cable Modem Termination Systems when supporting policy based
         service flows."
    ::= { docsQosGroups 4 }

docsQosServiceClassGroup OBJECT-GROUP
    OBJECTS {
    docsQosServiceClassStatus,
    docsQosServiceClassPriority,
    docsQosServiceClassMaxTrafficRate,
    docsQosServiceClassMaxTrafficBurst,
    docsQosServiceClassMinReservedRate,
```

```
    docsQosServiceClassMinReservedPkt,
    docsQosServiceClassMaxConcatBurst,
    docsQosServiceClassNomPollInterval,
    docsQosServiceClassTolPollJitter,
    docsQosServiceClassUnsolicitGrantSize,
    docsQosServiceClassNomGrantInterval,
    docsQosServiceClassTolGrantJitter,
    docsQosServiceClassGrantsPerInterval,
    docsQosServiceClassMaxLatency,
    docsQosServiceClassActiveTimeout,
    docsQosServiceClassAdmittedTimeout,
    docsQosServiceClassSchedulingType,
    docsQosServiceClassRequestPolicy,
    docsQosServiceClassTosAndMask,
    docsQosServiceClassTosOrMask,
    docsQosServiceClassDirection
    }

    STATUS  current
    DESCRIPTION
        "The docsQosServiceClassTable objects. If a CMTS implements
         expansion of Service Class Names in a QOS Parameter Set,
         this group is mandatory on the CMTS. If the CMTS does not
         support Service Class Names, this group may be unimplemented
         in the CMTS. This group is not implemented on the CM.
        "
    ::= { docsQosGroups 5 }
```

Appendix I Business Process Scenarios For Subscriber Account Management (informative)

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

I.1 The old service model: “one class only” and “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 that all IP packets have equal right-of-way over the Internet, it is a “one class fits all”, “first-come, first-served” 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.

I.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 would be 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 enough pressure on the finite supply of water and electricity that most of your normal demands for usage would likely go unfulfilled.

I.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.

I.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 first 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 second 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 third requirement is to capture a subscriber's actual usage, calculating the bill based on the rate associated with the customer's subscribed service levels.

I.3.2 Designing Classes 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 classes of service:

- "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.

I.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.

I.3.4 Designing Usage-Based Billing Models

In support of the offering of different classes of service 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.

This page intentionally left blank.

Appendix II Summary of CM Authentication and Code File Authentication (informative)

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

II.1 Authentication of the DOCSIS 2.0-compliant CM

If the CMTS is DOCSIS 2.0/BPI+-compliant and a DOCSIS 2.0 CM is provisioned to run BPI+ by the configuration file, the CMTS authenticates the CM by verifying the CM certificate and the manufacturer certificate. These certificates are contained in the Auth request and Auth Info packets respectively, and are sent to the CMTS by the CM after registration (when provisioned to do so by the configuration file). Only CMs with valid certificates will be authorized by the CMTS. Note that this CM authentication will not be applied if the CMTS and/or the CM is not compliant with BPI+, or the CM is not provisioned to run BPI+.

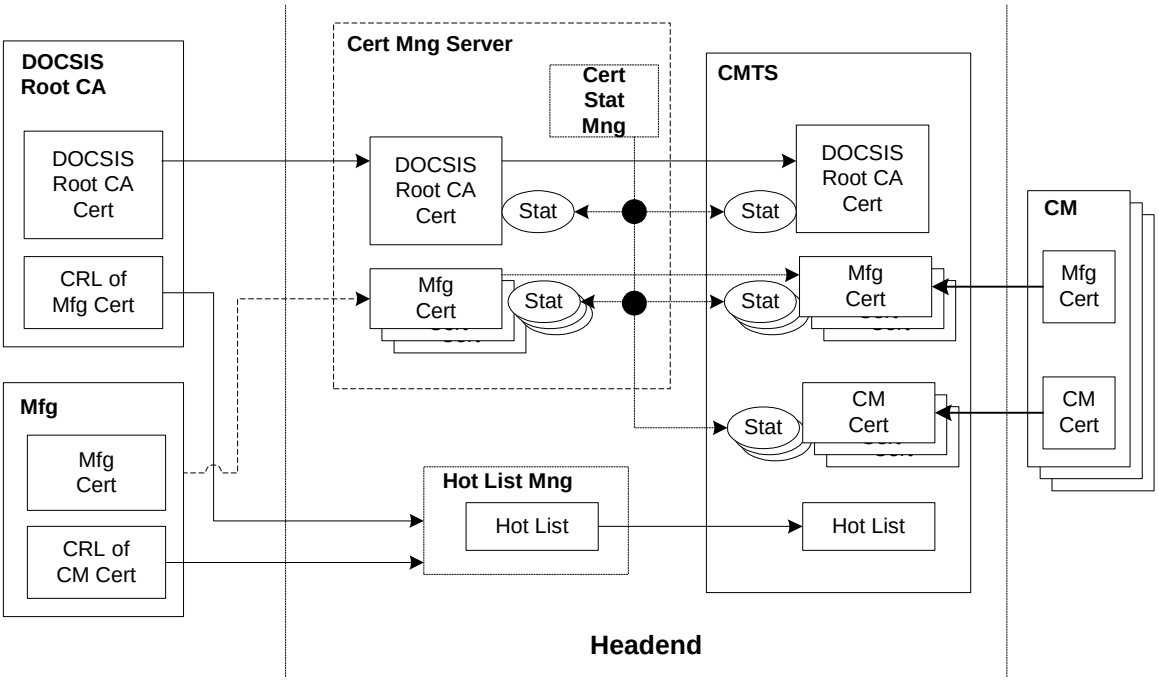


Figure II-1 Authentication of the DOCSIS 2.0-compliant CM

II.1.1 Responsibility of the DOCSIS Root CA

The DOCSIS Root CA is responsible for the following:

- Storing the DOCSIS Root private key in secret
- Maintaining the DOCSIS Root CA certificate
- Issuing the manufacturer CA certificates signed by the DOCSIS Root CA
- Maintaining the CRL of the manufacturer CA
- Providing the operators with the CRL

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.

II.1.2 Responsibility of the CM manufacturers

The CM manufacturers are responsible for the following:

- Storing the manufacturer CA private key in secret
- Maintaining 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.
- Issuing the CM certificates
- Putting the manufacturer CA certificate in the CM's software
- Putting each CM certificate in the CM's permanent, write-once memory
- Providing the operators with the hot list of the CM certificates. The hot list may be in CRL format. However, the detail of the format and the way of delivery are TBD.

II.1.3 Responsibility of the operators

The operators are responsible for the following:

- Maintaining that the CMTSes 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.
- Putting the DOCSIS Root CA certificate in the CMTS during the CMTS provisioning using the BPI+ MIB or the CMTS's proprietary function. The operator may have a server to manage this certificate for one or more CMTS(s).
- Putting the manufacturer CA certificate(s) in the CMTS during the CMTS provisioning using the BPI+ MIB or the CMTS's proprietary function (optional). The operator may have a server to manage this certificate for one or more CMTSes.
- Maintaining the status of the certificates in the CMTSes if desired using the BPI+ 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 CMTSes.

  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 CMTSes.
- Maintaining 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.

II.2 Authentication of the code file for the DOCSIS 2.0-compliant CM

When a DOCSIS 2.0/BPI+-compliant CM downloads a code file from a TFTP server, the CM must authenticate the code file as defined in Appendix D of the BPI+ specification regardless of whether the CM was provisioned to use BPI+, BPI, or neither, by the configuration file. The CM installs the new image and restarts using it only if verification of the code image was successful (as defined in Appendix D of the BPI+ specification). If authentication fails, the CM rejects the code file downloaded from the TFTP server and continues to operate using the current code. The CM performs a software download, whether initiated by the configuration file or SNMP, only if it was initialized with a valid CVC received 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.

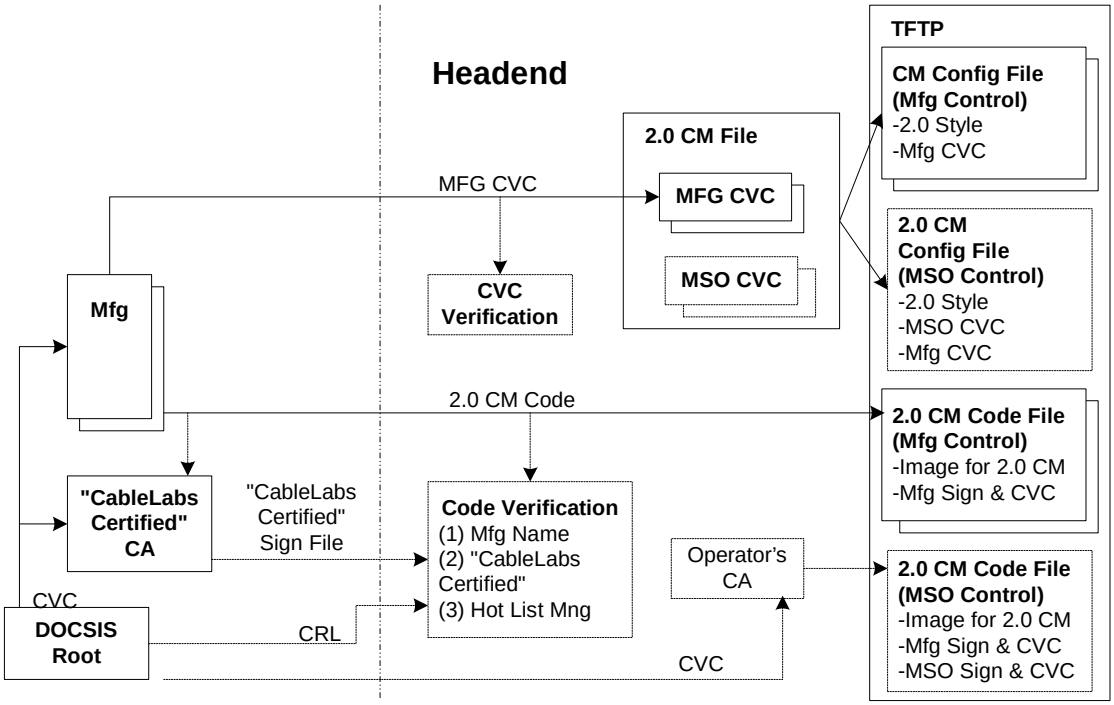


Figure II-2 Authentication of the code file for the DOCSIS 2.0-compliant CM

II.2.1 Responsibility of the DOCSIS Root CA

The DOCSIS Root CA is responsible for the following:

- Storing the DOCSIS Root private key in secret
- Maintaining the DOCSIS Root CA certificate
- Issuing the code verification certificates (CVCs) for the CM manufacturers, for the operators, and for "CableLabs Certified(TM)"
- The Root CA may maintain the CRL of the CVCs and provide it to the operators.

II.2.2 Responsibility of the CM manufacturer

The CM manufacturers are responsible for the following:

- Storing the manufacturer CVC private key in secret
- Putting the DOCSIS Root CA certificate in the CM's software
- Maintaining the manufacturer CVC (the current BPI+ specification only allows CVCs signed by the DOCSIS Root CA and does not accept self-signed CVCs)
- Generating the code file with the manufacturer's CVC and signature
- Providing the operators with the code file and the manufacturer CVC

II.2.3 Responsibility of CableLabs

CableLabs is responsible for the following:

- Storing the "CableLabs Certified(TM)" CVC private key in secret
- Maintaining the "CableLabs Certified(TM)" CVC signed by the DOCSIS Root CA
- Issuing the "CableLabs Certified(TM)" signature file for the DOCSIS 2.0 CM code file certified by CableLabs

II.2.4 Responsibility of the operators

Operators have the following responsibilities and options:

- Verifying the manufacturer CVC and signature in the code file provided by the manufacturer prior to using it (optional). The code file may be rejected (not used to upgrade CMs) if the manufacturer signature or CVC is invalid.
- Checking if the code file provided by the CM manufacturer is "CableLabs Certified(TM)" by verifying the "CableLabs Certified(TM)" CVC and signature in the "CableLabs Certified(TM)" signature file against the code file before the operator loads the code file on the TFTP server (optional).
- Maintaining 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)
- Generating the MSO-controlled code file by adding the operator's CVC and signature to the original code file provided by the CM manufacturer (optional)
- Checking if the CVC provided by the CM manufacturer is valid (optional)
- Putting the appropriate CVC(s) in the CM configuration file. In the 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 a CVC is not present the CM configuration file, or the CVCs that are present are invalid, the CM will not initiate a software download if instructed to via SNMP or the CM configuration file. Note that the DOCSIS 2.0-compliant CM may be registered and authorized by the CMTS and become operational regardless of whether the CM configuration file contains valid CVCs.

Appendix III Acknowledgments (informative)

On behalf of CableLabs I would like to thank the following key contributors to DOCSIS 2.0 for their outstanding and superb contributions to this valuable program…

Victor Hou of Juniper Networks (formerly Pacific Broadband) and Yoav Hebron of Conexant led the Physical Layer working groups that rewrote RFI Section 6, and wrote RFI Appendix VII. Ariel Yagil of Texas Instruments, Mike Grimwood of Imedia, Bruce Currivan and Tom Kolze of Broadcom, Hikmet Sari and David Munro of Juniper Networks, David Hull and Shimon Tzukerman of Conexant, Elias Nemer and Hassan Yaghoobi of Intel, and Jack Moran of Motorola participated in those groups.

John Chapman and Dan Crocker of Cisco led the DMPI working group that developed the specification for the CMTS MAC/PHY interface, which became RFI Annex H.

Rich Prodan of Terayon led the OSS working group that developed the new MIB for DOCSIS 2.0 as well as reworking the OSSI specification. Aviv Goren of Terayon, David Raftus of Imedia, Greg Nakanishi of Motorola, Adi Shaliv of Intel, Rich Woundy of Cisco, and Jason Schnitzer of Stargus contributed to that group.

Rusty Cashman of Correlant led the MAC layer working group that reworked much of RFI Sections 8, 9, and 11. Jeff Hoffman of Intel; Lisa Denney of Broadcom; Alon Bernstein of Cisco; Gordon Li of Conexant; Asaf Matatyaou of Terayon; Robert Fanfelle of Imedia; David Doan, Christiaan Prins, Leo Zimmerman and Simon Brand of Philips contributed to that working group.

Clive Holborow of Motorola led the System Capabilities working group which rewrote RFI Section 3 and Annex G, and contributed to RFI Sections 6, 9, and 11. Daniel Howard of Broadcom, Noam Geri of TI, and Doug Jones of YAS contributed to that group.

Clive Holborow of Motorola, Victor Hou of Juniper Networks, Mike Grimwood of Imedia, Bruce Currivan and Daniel Howard of Broadcom, Rich Prodan of Terayon, and Hal Roberts of ADC wrote the informational material in RFI Appendix VIII.

David Hull of Conexant, Luc Martens and Wim De Ketelaere of tComLabs, the engineers at UPC, and the Euro-DOCSIS Certification Board for their contributions to RFI Annex F.

The engineers at Terayon, Imedia, Broadcom, Texas Instruments, and Conexant, as well as the members of the IEEE 802.14a Hi-PHY working group (chaired by Roger Durant of Cabletron (now Riverstone)) developed the technology proposals that became DOCSIS 2.0.

George Hart of Rogers Cable, Oleh Sniezko of AT&T Broadband, Dan Rice of Stargus for their guidance and contributions on behalf of CableLabs member companies.

I would also like to recognize Greg White, Mukta Kar, John Eng, Doug Jones, Eduardo Cardona, Dorothy Raymond, Alex Ball, and Cynthia Metsker from CableLabs for their leadership and first class work.

CableLabs and the cable industry as a whole are grateful to these individuals and organizations for their outstanding, first class contributions.

Rouzbeh Yassini
CEO of YAS ventures, LLC
Exec Consultant to CableLabs

This page intentionally left blank.

Appendix IV Revisions (informative)

IV.1 ECNs included in SP-OSSIv2.0-I02-020617

Table IV-1 Incorporated ECN Table for SP-OSSIv2.0-I02-020617

ECN	Date Accepted	Summary
ossi2-n-02016	02/27/02	Replacement of RF MIB, associated changes.
oss2-n-02024	03/06/02	Version 2.0-specific items.
oss2-n-02053	04/10/02	Delete section 7.3.3.1.
oss2-n-02054	04/03/02	Replace section 7.3.4.1.
oss2-n-02062	04/10/02	Change section 7.4.2.1 and Annex F.
oss2-n-02069	05/01/02	Update to latest IPDR.org specifications.
oss2-n-02079	05/22/02	Clarify 'Storage type' functionality within CMs.
oss2-n-02088	05/29/02	Simplification of the specification.
oss2-n-02107	05/22/02	Update RFI MIB.

IV.2 ECNs included in SP-OSSIv2.0-I03-021218

Table IV-2 Incorporated ECN Table for SP-OSSIv2.0-I03-021218

ECN	Date Accepted	Summary
oss2-n-02087	06/12/02	Require the CMTS to support the docsDevBaseGroup.
oss2-n-02120	07/03/02	Correct value of ifSpeed for a USB 1.1 interface.
oss2-n-02124	07/03/02	Clarify wording of two event descriptions.
oss2-n-02134	07/31/02	Correct errors in table 7-6 regarding accessibility of the IF-EXT MIB.
oss2-n-02147	08/14/02	Specify the status of VACM entries in section 5.2.4.
oss2-n-02149	08/14/02	Make RFC2665 error counters optional for CMs and CMTSes.
oss2-n-02151	08/14/02	Make adjustments in Annex A.
oss2-n-02154	08/14/02	Raise priority of three DCC events from Error to Critical.
oss2-n-02158	08/14/02	Better define requirements in section 9.3, CM Diagnostic Capabilities.
oss2-n-02159	08/14/02	Redefine requirements for information available before CM registration.
oss2-n-02160	08/14/02	Augment the number of rows of VACM TreeFamily entries required.
oss2-n-02172	08/14/02	Clarify application of IP, LLC, and NmAccess filters apply to IfIndexes.
oss2-n-02175	09/18/02	Update the Event Messages E206.0 through E209.0.
oss2-n-02191	11/13/02	Correct terms and typo errors, fix discrepancies between v1.1 and v2.0.
oss2-n-02199	11/20/02	Allow use of multicast SIDs in the docsBpiCmTEKTable.
oss2-n-02205	11/06/02	Change the validity periods of BA Mfg and Co-signer CVCs.
oss2-n-02212	11/27/02	Provide new architecture guidelines for interfaces management.
oss2-n-02220	11/27/02	Update DOCS-QOS-MIB related MIB objects' default values.

IV.3 ECNs included in SP-OSSIv2.0-I04-030730

Table IV-3 Incorporated ECN Table for SP-OSSIv2.0-I04-030730

ECN	Date Accepted	Summary
OSS2-N-02193	12/04/02	Update section7.4 of Spec to lineup OSSI v1.1 spec and Clarify the Event priority requirements for a safe access to local log From CPE
OSS2-N-02221	12/04/02	Correct DOCS-IF-MIB docsIfQosProfMaxTransmitBurst range according to SP-RFIv2.0-I02-020617
OSS2-N-02234	01/02/03	Changing the specification reference to a new version of the BPI+ specification and the BPI+ MIB draft
OSS2-N-02235	01/02/03	Add a missing event code for DCC.
OSS2-N-02236	01/02/03	Wholesale replacement of section 7.1 and Appendix B to include new spec language throughout, new IPDR schema to support CMdocsisMode element and reporting of DOCSIS 1.0 services, and secure user authentication and secure file transfer via SSH2/SFTP
OSS2-N-03010	02/12/03	ECNs OSS-N-02174 and OSS2-N-02175 made changes to Event Log messaging and inadvertently removed too much of the text.
OSS2-N-03014	02/19/03	Correct an inconsistency regarding access to SNMPv3 MIBs in two areas of the specification
OSS2-N-03017	03/05/03	This ECN corrects a CM priority
OSS2-N-03021	03/12/03	Changing the specification reference to the draft-ietf-ipcdn-bpiplus-mib-05.
OSS2-N-03023	03/12/03	Update OSS specification to support draft-ietf-ipcdn-docs-rfmibv2-05, whose major new feature is the ability to provide CMTS upstream/downstream channel utilization statistics
OSS2-N-03025	6/13/03	Operators have shown interest in having a normalized CM front panel LEDs to simplify users self-diagnostic and customer support when recognizing the CM registration state.
OSS2-N-03034	4/9/03	This ECR corrects CM and CMTS default Priority
OSS2-N-03046	4/30/03	Allows CM devices the freedom to log Vendor-specific messaging for all Event Priorities in the local-log
OSS2-N-03053	5/29/03	This document proposes utilizing, as an option, the new key instead of the original key in the re-issued DOCSIS Manufacturer CVC
OSS2-N-03065	7/2/03	Update OSS specification to support draft-ietf-ipcdn-docs-rfmibv2-05 clarification on docsIfCmStatusValue.
OSS2-N-03069	7/2/03	Redefine docsIfCmtsCmStatusValue compared to the current MIB requirement draft-ietf-ipcdn-docs-rfmibv2-05.txt to support backward compatibility CMTS operations
OSS2-N-03071	7/2/03	Minor corrections to OSSI2 spec (Omnibus)

IV.4 ECNs included in SP-OSSIv2.0-I05-040407

Table IV-4 Incorporated ECN Table for SP-OSSIv2.0-I05-040407

ECN	Date Accepted	Summary
OSS2-N-03067	8/20/03	Update OSSI spec to comply with the new set of SNMP V3 standards.
OSS2-N-03083	8/6/03	Apply old OSS-N-02022 to 2.0 OSSI specification to allow vendor specific text to be added at the end of events.
OSS2-N-03087	9/3/03	Add an enum in DOCS-IF-MIB:docsIfCmtsModPreambleType for a row entry consisting only of DOCSIS 1.x bursts.
OSS2-N-03090	9/10/03	Corrections to OSSI 2.0 spec Annex A
OSS2-N-03091	9/1703	Correct the DOCS-IF-MIB:docsIfCmtsChannelUtId range in draft-ietf-ipcdn-docs-rfmibv2-05.txt.
OSS2-N-03092	11/12/03	Add guideline for the modulation profile especially for assigning modulation profile to upstream channels regarding the channel type.
OSSIv2.0-N-03.0108-2	12/03/03	To apply OSS-N-02060 to 2.0 Specifications.
OSSIv2.0-N-03.0112-2	1/7/04	Update the accuracy statement for Euro-DOCSIS implementations.
OSSIv2.0-N-03.0115-2	1/21/04	Accept duplicate TLV-11's when the values are the same
OSSIv2.0-N-03.0117-2	1/21/04	Update References for Syslog Formating
OSSIv2.0-N-04.0121-3	2/25/04	Allow more flexibility of modulation profiles implementation to meet the needs in fields.
OSSIv2.0-N-04.0126-6	3/10/04	Defines the MIB requirements in Section 6 pointing to the new Annex I "Requirements for DOCS-LOADBALANCING-MIB"
OSSIv2.0-N-04.0127-4	3/10/04	Clarifications for DCC in the QoS MIB
OSSIv2.0-N-04.0130-2	3/10/04	Clarification of notBefore value for re-issued CVC